11-24-2015

Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging

Dengyu Liu
dxl5849@rit.edu

Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging

by

Dengyu Liu

B.S. Beijing Institute of Technology, China, 2008
M.S. Beijing Institute of Technology, China, 2010

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

Nov 24, 2015

Signature of the Author _____

Accepted by _____
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Dengyu Liu
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

Dr. James A. Ferwerda, Thesis Advisor

Dr. Nathan Cahill

Dr. Gabriel Diaz

Dr. Reynold Bailey

Date

Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging

by

Dengyu Liu

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

Cameras face a fundamental tradeoff between spatial and temporal resolution. Digital still cameras can capture images with high spatial resolution, but most high-speed video cameras have relatively low spatial resolution. It is hard to overcome this tradeoff without incurring a significant increase in hardware costs. In this paper, we propose techniques for sampling, representing and reconstructing the space-time volume in order to overcome this tradeoff. Our approach has two important distinctions compared to previous works: (1) we achieve sparse representation of videos by learning an over-complete dictionary on video patches, and (2) we adhere to practical hardware constraints on sampling schemes imposed by architectures of current image sensors, which means that our sampling function can be implemented on CMOS image sensors with modified control units in the future. We evaluate components of our approach - sampling function and sparse representation by comparing them to several existing approaches. We also implement a prototype imaging system with pixel-wise coded exposure control using a Liquid Crystal on Silicon (LCoS) device. System characteristics such as field of view, Modulation Transfer Function (MTF) are evaluated for our imaging system. Both simulations and experiments on a wide range of scenes show that our method can effectively reconstruct a video from a single coded image while maintaining high spatial resolution.

Acknowledgements

First, I would love to thank my advisor Dr. James A. Ferwerda and Dr. Jinwei Gu for their continious support on my Master study and research. Dr. Ferwerda provides me continious support for my research and my life. When I had trouble with my research, Dr. Ferwerda gave me valuable advices to help me transit my program. Dr. Ferwerda give me not only advices on research, but also guidance on how to live a wonderful life. Dr. Gu leads me to the new area of computer vision and computational photography. Without his enthusiasm and encouragement, I cannot achieve such progress on my research. I would also like to thank my committees: Dr. Nathan Cahill, Dr. Gabriel Diaz, Dr. Reynold Bailey for their insightful comments and suggestions on my thesis.

Second, I would make a special thanks to Dr. Stefi Baum for her continious support on my study, which helps me to proceed my thesis smoothly. I want to thanks to Val Hemink for arranging MCSL a lovely place to stay. I really enjoyed my stay in MCSL for the first year of my research. And I also want to thank Beth Lockwood, Susan Chan, Joyce French for taking care of all the issues that students encounter. My gratitude also goes to National Science Foundation (NSF) and Xerox for providing financial support for my research.

Then I want to thanks my collaborator Yasunobu Hitomi, Yilong Liang, Mohit Gupta and Hajime Nagahara. Without your contribution, I would not be able to finish my thesis. I also want to thank my classmates Yang Hu, Chao Liu, Jun Jiang for helpful discussion on my research project. Thanks to my classmates Jiashu Zhang, Dong Wang, Ming Zhang, Bo Ding for sharing wonderful vacation time together.

Finally, I would like give my sincere gratitude to my parents, my grandma and my wife Weiwei. Thanks to my parents for taking me to this wonderful world, thanks to my grandma for taking care of me during my childhood, and special thanks to my wife, whom I met in India, for arranging a clean, cozy home everyday. Thank you all for your unconditional support.

To My Wife and Parents

Contents

List of Tables

List of Figures

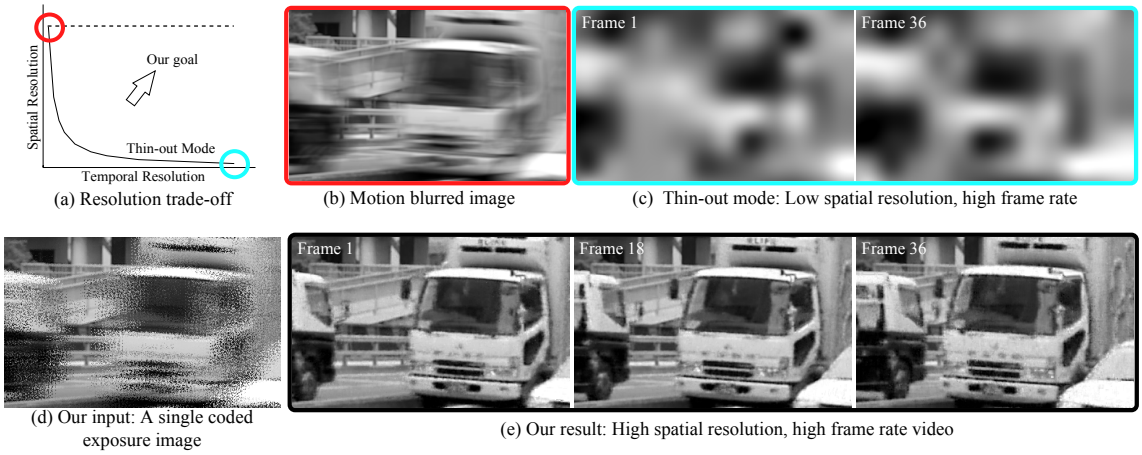
CHAPTER 1

Introduction

1.1 Motivations

Digital cameras are limited by a fundamental tradeoff between spatial resolution and temporal resolution. As the frame rate increases, the spatial resolution decreases. This limitation is caused by hardware factors such as the readout and Analog-to-Digital (A/D) conversion time of image sensors. Although it is possible to increase the readout throughput by introducing parallel A/D converters and frame buffers [21], it often requires more transistors per pixel and thus lowers the fill factor, reduces light efficiency and increases cost. As a compromise, many camera manufactures implement a "thin-out" mode (*i.e.*, high speed draft mode [3]), which directly trade off the spatial resolution for a higher temporal resolution, and thus degrades the image quality, as shown in Figure. 1.1. Can we go beyond this fundamental limitation and capture videos more efficiently?

Tracing back of the history of digital cameras, we find that the technology of digital cameras has developed significantly in the past few decades. From the first 100×100 CCD camera introduced by Fairchild[1], to the 40 Megapixels digital SLR camera, the resolution



(a) Resolution trade-off (b) Motion blurred image (c) Thin-out mode: Low spatial resolution, high frame rate

(d) Our input: A single coded exposure image (e) Our result: High spatial resolution, high frame rate video

Figure 1.1: Overcoming the space-time resolution tradeoff. (a) Digital cameras face a fundamental tradeoff between spatial resolution and temporal resolution. (b) A digital still camera has high spatial resolution but low temporal resolution, which often results in motion blur. (c) The "Thin-out" mode trades off the spatial resolution to increase the frame rate. For large frame rates, the image quality is severely degraded. (d) By capturing a pixel-wise coded exposure image, and learning a sparse representation of videos, (e) we achieve a high-spatial resolution and high frame rate video simultaneously.

and quality of the image sensor have been greatly enhanced. However, the underlying camera model is essentially the same as the conventional cameras: the conventional camera has film (similar to image sensors in digital camera) and a lens, applies a simple and restrictive sampling scheme on the complete set of rays or light fields that resides in the real scene.

In recent years, Nayar [27] proposed the concept of *computational camera*. The conventional camera follows the "camera obscura" principle and produces a linear perspective image. In contrast, the computational camera combines novel optics and computational modules, which encode and decode images to get new types of visual information such as the light field.

At the same time, a "big idea" in signal processing called Compressive Sensing (CS) was proposed[9, 13]. CS states that one can recover a signal from far fewer samples than that required by the Shannon-Nyquist sampling theorem. The recovery is guaranteed if the signal and the measurement meet certain requirements.

1.2 Objectives

In this thesis, with the knowledge of computational camera and CS, I will exploit the possibility to go beyond the fundamental limitation of digital camera and show its application in high speed imaging. As a result, the objectives for this thesis are:

- design a flexible space-time sampling scheme for video capturing, which adheres to the restrictions of existing hardware.
- propose an effective video capture and reconstruction scheme based on CS, which combines random sampling and sparse representation.
- implement a hardware prototype imaging system with pixel-wise coded exposure control using a Spatial Light Modulator (SLM).

1.3 Thesis Overview

The following chapters of the thesis are organized as follows:

Chapter 2 gives the background of CS and computational camera, followed by a literature review on related work.

Chapter 3 describes the methodology of flexible space-time video sampling and reconstruction.

Chapter 4 illustrates the hardware implementation of our pixel-wise coded exposure imaging system.

I will give a conclusion of our project with a summary and discussion for future research in Chapter 5.

CHAPTER 2

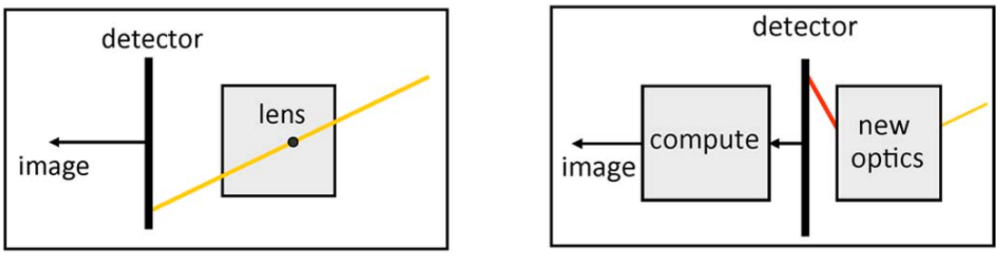
Background and Related Work

In this chapter, I will give an introduction on computational cameras with its definition and design approaches; compressive sensing with its theorem and algorithms. Related work will be discussed in the end.

2.1 Computational Cameras

As shown in Figure. 2.1, a traditional camera follows the basic principle of "camera obscura", which consists of a detector (film or sensor) and a lens to capture the light rays passing through its center of projection. It only captures a subset of the light fields. In contrast, a computational camera samples in a different way to obtain new forms of visual information. It adds new optics to code the images, and combines computational modules to decode the captured images to produce new types of images. Those new types of images can either be meaningful to a human observer or a computer for scene interpreting.

Zhou and Nayar [51] summarized six coding approaches that are used in the optical design of computational cameras, as shown in Figure. 2.2.



(a) Traditional Camera (b) Computational Camera

Figure 2.1: Traditional cameras follow the principle of "camera obscura", whereas computational camera add new optics and computational modules to modulate the light and get new information from the scene.[29]

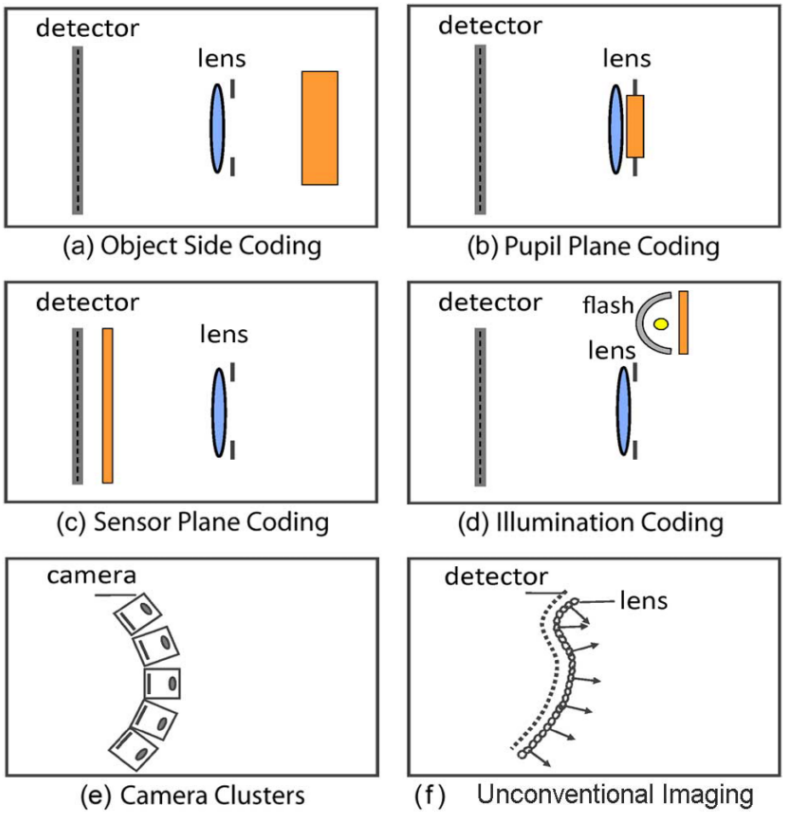


Figure 2.2: Optical Coding approaches[51]

- **Object side coding** is the most convenient way to implement. It places the mask directly in front of the lens, providing spatially varying light modulation. Applications of light field imaging, depth estimation, High Dynamic Range (HDR) imaging often use this coding scheme.
- **Pupil side coding** puts the coded mask in the aperture plane of the lens. Since all the light rays pass the same pupil plane, pupil side coding provides spatially invariant coding and modulates the point spread function of the system. It is usually applied in applications such as light field, extended depth of field.
- **Sensor side coding** locates the mask either on the same plane of the sensor or close to the sensor plane. The mask on the sensor plane achieves a pixel-wise modulation of the sensor plane, while the mask in front of the sensor modulates the light both spatially and angularly. This coding scheme can be applied to applications such as light field, HDR imaging and high speed imaging.
- **Illumination coding** modulates the captured images by using a spatially/temporally controllable light source. The light source can be a camera flash or a projector. This technique is widely used for 3D reconstruction, depth estimation etc.
- **Camera clusters** combines multiple cameras to obtain a more flexible way to overcome the limits of individual cameras. It is often used in the fields of high speed imaging, HDR imaging, synthetic aperture etc.
- **Unconventional coding** consists computational cameras that use unconventional sensor architectures such as micro lens arrays to obtain new information. One application using micro lens arrays is to capture light fields.

2.2 Compressive Sensing

Most of the existing data acquisition systems follow the classical Shannon/Nyquist sampling theorem, i.e., the sampling rate must be at least twice of the maximum frequency of

the signal so as to avoid losing information. In many applications such as digital imaging, the Nyquist rate is so high that compression is necessary for storage and transmission.

2.2.1 Sparsity and Transform Coding

It is recognized that many natural signals are *sparse* or *compressible* in a convenient basis. Consider a one-dimensional, discrete-time signal $\mathbf{x} \in \mathbb{R}^N$, which is an $N \times 1$ column vector. It can be represented as a linear combination of a series of basis functions $\mathbf{D} = \{d_i\}_{i=1}^N$:

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}, \tag{2.1}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^N$, and α_i is the weighting coefficient of d_i, which can be calculated as $\alpha_i = \langle \mathbf{x}, d_i \rangle$. Then $\boldsymbol{\alpha}$ is the equivalent representation of $\mathbf{x}$ in $\mathbf{D}$ domain.

The signal $\mathbf{x}$ is *K-sparse* when only K of the coefficients are nonzero. The signal is *compressible* when $\mathbf{x}$ has only a few large coefficients and many small ones.

The conventional approach for compression is transform coding. A compressible signal can be well approximated by its K-sparse representation. The strategy for compression and decompression can be described as follows:

> Encoding: Construct $\mathbf{D}$, do transform coding $\boldsymbol{\alpha} = \mathbf{D}^T x$, keep the value and locations of the K largest coefficients in $\boldsymbol{\alpha}$
>
> Decoding: Put back those K coefficients back to original locations, put zeros in other locations to form $\hat{\boldsymbol{\alpha}}$, construct $\mathbf{D}^{-1}$, do inverse transform to reconstruct $\hat{\mathbf{x}} = \mathbf{D}^{-1}\hat{\boldsymbol{\alpha}}$

This *sample-then-compress* framework is actually inefficient. First, considering the Shannon's theorem, in order to get a better resolution of the signal, the initial number of samples N may be quite large even if the actual K is small. Second, all the N coefficients $\boldsymbol{\alpha}$ need to be computed, even though all but K of them will be discarded. Third, the locations of the K largest coefficients depend on the signal itself, thus this strategy is *adaptive*. In addition, extra memory is needed to store those information.

2.2.2 Compressed Sampling

Aiming to solve the above inefficiency issues, CS theory says that one can directly acquires a compressed signal without going through the intermediate stage of sampling all N samples. For a signal $\mathbf{x} \in \mathbb{R}^N$, consider a linear measurement matrix $\mathbf{S}$ applied to the signal $\mathbf{x}$, *i.e.*, $\mathbf{y} = \mathbf{Sx}$, $\mathbf{S} \in \mathbb{R}^{M \times N}$. Here, each row of $\mathbf{S}$ is a sensor, which is multiplied with the signal, get an acquisition of part of the signal. Combined with the signal representation in Equation (2.1), $\mathbf{y}$ can be written as:

$$\mathbf{y} = \mathbf{Sx} = \mathbf{SD}\boldsymbol{\alpha} = \boldsymbol{\Theta}\boldsymbol{\alpha}, \tag{2.2}$$

where $\boldsymbol{\Theta} = \mathbf{SD}$ is an $M \times N$ sensing matrix. The measurement process is *non-adaptive*, which means that the rows of $\mathbf{S}$ are fixed and are not related to the signal $\mathbf{x}$. To solve Equation (2.2), we need to first design a stable measurement matrix $\mathbf{S}$ such that the essential information in the signal is not damaged by the dimensionality reduction, then develop an algorithm to recover signal $\mathbf{x}$ from $M \ll N$ measurements.

Since $M \ll N$, the problem is ill-conditioned. A necessary and sufficient condition to make this problem well-conditioned is that the sensing matrix $\boldsymbol{\Theta}$ satisfy the Restricted Isometry Property (RIP)[10], *i.e.*, for each integer $k = 1, 2, \cdots$, define the isometry constant $\delta_k \in (0, 1)$, such that

$$(1 - \delta_k) \parallel \mathbf{x} \parallel_2^2 \leq \parallel \boldsymbol{\Theta}\mathbf{x} \parallel_2^2 \leq (1 + \delta_k) \parallel \mathbf{x} \parallel_2^2 . \tag{2.3}$$

When this property holds, $\boldsymbol{\Theta}$ approximately preserves the energy of K-sparse signals. An equivalent way to describe the RIP is to say that all subsets of K columns taken from $\boldsymbol{\Theta}$ are nearly orthogonal. It is also proven that at least $M \gg Klog(N/K)$ measurements are necessary to achieve the RIP.

While the RIP provide guarantees for the recovery of K-sparse signals, verifying that a matrix $\boldsymbol{\Theta}$ satisfies the RIP has a combinatorial computational complexity of $\binom{n}{k}$ combinations. Therefore, it is preferable to use other properties of $\boldsymbol{\Theta}$ that are easily computable to provide more concrete recovery guarantees. One condition is called *incoherence*, which

requires that for $\mathbf{\Theta} = \mathbf{SD}$, the rows of $\mathbf{D}$ cannot sparsely represent the columns of $\mathbf{S}$.

It is found that a random matrix, *e.g.*, with independent and identically distributed (i.i.d) random variables from a Gaussian probability density function , can satisfy both the RIP and *incoherence*. Using a random matrix has several benefits. First, because the random measurements are *democratic*, it is more robust to the loss or corruption of a fragment of measurements. Second, if $\mathbf{D}$ is an orthonormal basis, with a Gaussian distribution measurement matrix $\mathbf{S}$, the sensing matrix $\mathbf{\Theta} = \mathbf{SD}$ is also Gaussian, thus $\mathbf{\Theta}$ will satisfy the RIP with a high probability. This property is referred as *universality*.

2.2.3 Signal Reconstruction

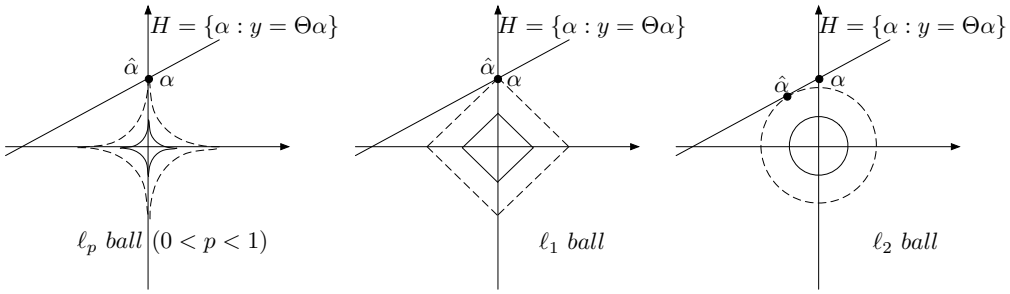


Figure 2.3: Minimization comparison with different norms

The signal reconstruction algorithm takes the M measurements of $\mathbf{y}$, and the sensing matrix $\mathbf{\Theta}$ as inputs, to reconstruct the signal x. Since $M \ll N$, this is an under determined system, and there are infinitely many solutions. The traditional approach to get a unique solution is using least square regression (or ℓ_2 norm minimization) by solving

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_2 \quad \text{subject to } \mathbf{y} = \mathbf{\Theta}\boldsymbol{\alpha}. \tag{2.4}$$

This optimization has the convenient closed-form solution $\hat{\boldsymbol{\alpha}} = \mathbf{\Theta}^T(\mathbf{\Theta}\mathbf{\Theta}^T)^{-1}\mathbf{y}$. However, ℓ_2 norm minimization almost never find a K-sparse solution, as explained in Figure 2.3. It can only get a solution with many nonzero elements.

ℓ_0 norm counts the number of nonzero entries in $\boldsymbol{\alpha}$, thus may be used to optimize the problem. ℓ_0 norm minimization solves the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to } \mathbf{y} = \boldsymbol{\Theta}\boldsymbol{\alpha}. \tag{2.5}$$

Unfortunately, the objective function $\|\cdot\|_0$ is nonconvex, and Equation (2.5) is NP-hard. One avenue to make this problem more tractable is to replace $\|\cdot\|_0$ with its convex approximation $\|\cdot\|_1$:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{subject to } \mathbf{y} = \boldsymbol{\Theta}\boldsymbol{\alpha}. \tag{2.6}$$

This non-linear convex optimization problem can be reduced as a linear program, and solved by basis pursuit[11].

While convex optimization are powerful methods for solving sparse representation problem, there are also greedy approaches which usually are more time-efficiency. Greedy algorithms rely on an iterative approximation of the signal coefficients and support, by obtaining an improved estimate of the sparse signal at each iteration until a convergence criterion is met. One simple and popular approach is Orthogonal Matching Pursuit (OMP)[23]. It is different from matching pursuit in that the residual is always orthogonal to the atoms already selected. This means that the same atom will never be re-selected and leads to a faster convergence.

Algorithm 1 Orthogonal Matching Pursuit

1: Input: basis $\mathbf{D}$, signal $\mathbf{x}$, target sparsity K or target error ε
2: Output: Sparse representation $\boldsymbol{\alpha}$ such that $\mathbf{x} \approx \mathbf{D}\boldsymbol{\alpha}$
3: Initialize: Set $\mathbf{I} := (), \mathbf{r} := \mathbf{x}, \boldsymbol{\alpha} := 0$
4: **while** (*stopping criterion not met*) **do**
5: Find $d_i \in \mathbf{D}$ with maximum inner product $|\langle \mathbf{r}, d_i \rangle|$
6: $\mathbf{I} := [\mathbf{I} \quad d_i]$
7: Get approximation of $\hat{\mathbf{x}}$ by least square minimization: $\hat{\mathbf{x}} := (\mathbf{D})^{+}\boldsymbol{\alpha}$
8: Update residual $\mathbf{r}$ with $\mathbf{r} := \mathbf{x} - \mathbf{D}\boldsymbol{\alpha}$

2.2.4 Sparse representation via learning

Compressive sensing prefers that the signal is sparse in a proper basis or dictionary. The overcomplete dictionary that leads to a sparse representation can be chosen as a predefined set of functions. Overcomplete dictionaries such as wavelets, curvelets and Fourier transform have been applied to signal/image compression applications. The predefined dictionary is appealing because of its simplicity. The success of these dictionaries depends on how well the signal is represented sparsely.

Another approach to design an overcomplete dictionary is by adapting its content to fit a given set of signal examples. The goal is that the learned dictionary yields a sparse representation of the training signal, which outperforms the pre-determined dictionaries. The dictionary learning approach can be formalized as the following optimization problem:

$$\arg\min_{\mathbf{D},\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_F^2 \quad \text{subject to } \|\boldsymbol{\alpha}\|_0 \leq K, \tag{2.7}$$

where $\|\cdot\|_F$ is the Frobenius norm. This role of the penalty and constraints in Equation (2.7) can also be reversed:

$$\arg\min_{\mathbf{D},\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to } \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_F^2 \leq \epsilon. \tag{2.8}$$

The problem can be solved using an alternative minimization technique, and can be treated as a generalized K-Means. The first step finds the coefficients given the dictionary, which is the *sparse coding stage*. Then the dictionary is updated with fixed coefficients, which is the *dictionary update stage*[14]. Different dictionary design algorithms vary in the calculation of coefficients and update of dictionary. Olshausen and Field [31] constructs the dictionary from a probabilistic perspective. The dictionary is constructed by solving a Maximum Likelihood estimation:

$$\mathbf{D} = \arg\max_{\mathbf{D}} P(\mathbf{x}, \alpha|\mathbf{D}) \tag{2.9}$$

$$= \arg\min_{\mathbf{D}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}|^2 + \lambda\|\boldsymbol{\alpha}\|_1\}. \tag{2.10}$$

An iterative method was used to solve Equation (2.10) for both sparse coding and dictionary update.

However, the iterative update approach can be slow. Engan et al. [15] proposed another dictionary learning algorithm called Method of Optimal Direction (MOD). The main contribution of MOD is its simple and efficient implementation for dictionary update. In the sparse coding stage, a pursuit algorithm is used to get the coefficients. In the dictionary update stage, they solve for dictionary $\mathbf{D}$ by least-squares:

$$\mathbf{D} = \arg\min_{\mathbf{D}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_F^2 \tag{2.11}$$

$$= \mathbf{x}\boldsymbol{\alpha}^T(\boldsymbol{\alpha}\boldsymbol{\alpha}^T)^{-1} \tag{2.12}$$

$$= \mathbf{x}\boldsymbol{\alpha}^+, \tag{2.13}$$

Aharon et al. [5] proposed a dictionary learning method called K-SVD. It follows a similar scheme of MOD, but uses a different method to update the dictionary. Instead of updating the dictionary at one time, in K-SVD, the dictionary $\mathbf{D}$ are processed in atoms (i.e., columns) sequentially. In each step, only the signal $\mathbf{x}$ whose sparse representation uses the current atom is updated, while the other atoms are fixed.

$$\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_F^2 = \left\|\mathbf{x} - \sum_{i=1}^{m} d_i \alpha_i^t\right\|_f^2 \tag{2.14}$$

$$= \left\|(\mathbf{x} - \sum_{i \neq j} d_i \alpha_i^T) - d_j \alpha_j^T\right\|_F^2 \tag{2.15}$$

$$= \left\|\mathbf{E}_j - d_j \alpha_j^T\right\|_f^2, \tag{2.16}$$

where d_i is the i-th column of $\mathbf{D}$, α_i is the i-th row of $\boldsymbol{\alpha}$, and E_j is an error matrix refer to the j-th dictionary atom. The minimization of d_j and α_j are rank-1 minimization tasks, which can be solved directly via an SVD decomposition.

Algorithm 2 K-SVD

1: Input: initial dictionary $\mathbf{D}_0$, signal $\mathbf{x}$, target sparsity K or target error ε
2: Output: Dictionary $\mathbf{D}$ and Sparse representation $\boldsymbol{\alpha}$ such that $\mathbf{x} \approx \mathbf{D}\boldsymbol{\alpha}$
3: Initialize: Set $D := D_0$ with ℓ_2 normalized columns
4: **for** j=1...m **do**
5: $\hat{\boldsymbol{\alpha}}_i = arg\min \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \quad subject\ to\ \|\boldsymbol{\alpha}\|_0 \leq K$ (Sparse Coding Stage)
6: **for** k=1...n **do** (Dictionary update stage)
7: ω_k :=indices of the signal in $\mathbf{x}$ whose representations use α_k
8: $\mathbf{E}_k := \mathbf{x} - \sum_{j \neq k} d_j \alpha_j^T$
9: Obtain $\mathbf{E}_k^R$ by shrinking $\mathbf{E}_k$ by choosing only the columns corresponding to ω_k
10: Apply SVD decomposition: $\mathbf{E}_k^R = U \Delta V^T$
11: Update dictionary column d_k

2.3 Related Work

2.3.1 Efficient Video Sampling and Representation

One way to achieve efficient video sampling is to design new sampling schemes. The coded global shutter (*i.e.*, flutter shutter) is the simplest approach, which has been used for motion deblurring [34] and reconstructing the periodic high speed motion with compressive

sensing [43]. Holloway et al. [20] also proposed a sampling scheme using flutter shutter, but instead of using a parametric motion model, they used video priors to reconstruct arbitrary videos. Llull et al. [22] mounted a pre-designed coded mask onto a piezoelectric stage. They moved the per-pixel mask during the integration time to achieve spatio-temporal modulation. Gu et al. [16] designed a coded rolling shutter for CMOS image sensors, simulation results shown that it can be applied for high speed imaging, high dynamic range imaging and so on. Portz et al. [33] proposed a coded sampling scheme assuming each pixel has a random permutation of exposure time and offset. Since there are no gaps between exposures, the sampling scheme maintains 100% light throughput. For each captured frame, only part of the pixels with varying exposures are sampled. By exploiting spatial and temporal redundancy, a high speed, high dynamic range video is reconstructed. Marcia et al. [25] proposed a method that applies coded aperture masks to video frames. Shu and Ahuja [41] proposed a circulant sampling scheme consists of random convolution and random subsampling, which reconstructs high spatial resolution videos from a low spatial resolution sensor.

There are also sophisticated sampling schemes using Spatial Light Modulators (SLMs) to achieve per-pixel modulation. Wakin et al. [44] used a Digital Micromirror Device (DMD) to build the single pixel camera for compressive video capturing using the sparsity of 3D wavelets basis. They made the assumption that each frame is static when capturing. However, this is usually not true for most scenes. In order to better deal with videos, Sankaranarayanan et al. [37] proposed a multi-scale video sensing and recovery framework for the single pixel camera. They designed a dual-scale sensing matrices which can generate a *preview* of the scene with low computational complexity. The motion information from the preview is estimated via optical flow, and then used for reconstruction. The DMD based single pixel camera is beneficial for imaging applications where building sensor arrays is impossible or the cost is extremely expensive such as infrared imaging. In other applications, Nayar et al. [29] proposed programmable imaging system using a DMD for HDR imaging, feature detection and object recognition. Ri et al. [36] also built a DMD camera to do phase analysis and shape measurement. Bub et al. [7] implemented

a pixel-wise coded exposure camera using a DMD for high speed imaging. They designed an optimized sampling function to let pixels expose at different subframes. Then they traded off the spatial resolution to obtain high speed videos by up sampling. Another popular SLM device called Liquid Crystal on Silicons (LCoS) is also widely used. Reddy et al. [35] proposed a programmable pixel-wise compressive camera based on LCoS. Since this technique relies on optical-flow based regularization, it cannot faithfully reconstruct scenes containing deforming objects, occlusion and specularity. And the exploited spatial redundancy is similar to traditional compression algorithms.

Efficient video sampling can also use multi-cameras system. Gupta et al. [17] proposed synthesizing high-resolution videos from low-resolution videos and a few high-resolution key frames. Ben-Ezra and Nayar [6] and Tai et al. [42] used a hybrid camera system to do motion-deblurring and temporal upsampling. Shechtman et al. [40] proposed an approach to combine multiple low resolution videos to reconstruct a high resolution video. They added a directional space-time regularization to constrain the solution. Wilburn et al. [47] built a dense camera array with an optimized timing control, and achieved a high-speed videography from interleaved exposures. However, in order to achieve high speed imaging, the exposure time is reduced. To increase the light throughput, Agrawal et al. [4] modified the multi-cameras system with a coded sampling. They also used CCD cameras instead of CMOS cameras to avoid *rolling-shutter* artifacts.

There are also adaptive methods to reconstruct videos. Gupta et al. [18] implemented a pixel-wise coded exposure imaging system using a projector-camera system. Their techniques make it possible to capture fast moving scene without motion blur, while simultaneously preserve a high spatial resolution of the static scene. Conventional compressive sensing techniques assume that there is an upper bound on the number of the significant coefficients in the signal, Warnell et al. [46] used the side information to predict the number of significant coefficients, and adaptively change the number of CS measurements for each image of the video sequence. Yang et al. [48] used a Gaussian Mixture Model (GMM) to describe the video patch. They adaptively changed the parameters of the GMM online, and also changed the temporal compression rate based on the complexity of the scene.

2.3.2 Video Reconstruction

Compressive sensing requires a reduced sampling rate, so the reconstruction is followed by seeking the sparse representation of the signal and exploiting the prior knowledge of the signal to constrain the solution. It is found that smooth images are sparse in the Fourier basis, and piecewise smooth images are sparse in the wavelet basis. The commercial coding standard of JPEG and JPEG 2000 exploit this sparsity[12]. In video CS, redundancy in the temporal and spatial domain are exploited for better reconstruction. For spatial redundancy, 2D/3D wavelets[20, 35, 37, 44, 50] basis are used as sparsity constrains, 2D total variation is able to reconstruct piece-wise constant images accurately and preserve the edge information in the image[20, 39]. For temporal redundancy, 3D total variation and optical flow are widely used to estimate motion and provide constrains for reconstruction[17, 18, 33, 35, 37]. Sankaranarayanan et al. [38] developed a framework for video CS to model the scene as a linear dynamical system. Marcia et al. [25] proposed an approach that minimizes the wavelet sparsity of the first frame and subsequent frame differences. Park and Wakin [32] proposed a multi-scale framework, which uses a coarse-to-fine reconstruction algorithm.

CHAPTER 3

Methodology

Natural images are generally smooth or piecewise smooth, thus they can be sparsely represented in a Fourier or wavelet basis. JPEG and JPEG2000 exploit this sparsity to do image compression. Olshausen and Field [31] exploited the sparsity of natural scenes from the perspective of visual perception. They modeled an Image as a linear combination of a series of basis functions. By solving an optimization problem with a sparsity constraint on the coefficients, they learned the basis functions which resemble the spatial property of the receptive field in simple cells. In another learning-based approach, Aharon et al. [5] proposed an algorithm to train overcomplete dictionaries for sparse representation. Compared with the pre-defined dictionaries such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), using learned dictionaries has better performance on image applications such as denoising.

In this thesis, we apply a similar approach of [5] to exploit statistical priors of time-varying appearance of natural scenes and propose a pixel-wise coded exposure to capture a video from a single photograph. Our key assumption is that the time-varying appearance of natural scenes can be represented as a sparse linear combination of the atoms of an

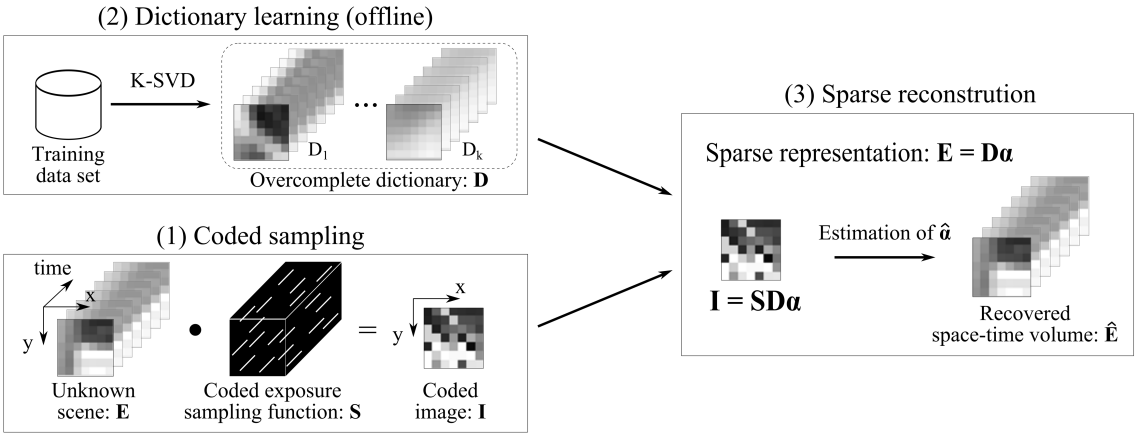


Figure 3.1: Overview of Our Approach. There are three main components of our approach: (1) coded exposure sampling and projection of space-time volumes into images, (2) learning an overcomplete dictionary from the training video data, and (3) sparse reconstruction of the captured space-time volume from a single coded image.

overcomplete dictionary learned from the training data. Thus, by using a pixel-wise coded exposure, we can obtain a 2D projection of the 3D space-time volume and reconstruct the volume via a sparse reconstruction algorithm.

3.1 Overview of Our Approach

Figure 3.1 shows the flow-chart of our approach. Let $E(x, y, t)$ denote the space-time volume corresponding to an $M \times M$ pixel neighborhood and one frame integration time of the camera. A conventional camera captures the projection of this volume along the time dimension, resulting in an $M \times M$ image patch. Suppose we wish to achieve an N times gain in temporal resolution, *i.e.*, we wish to recover the space-time volume E at a resolution of $M \times M \times N$. Let $S(x, y, t)$ denote the per-pixel shutter function of the camera within the integration time ($S(x, y, t) \in \{0, 1\}$). Then, the captured image $I(x, y)$ is

$$I(x, y) = \sum_{t=1}^{N} S(x, y, t) \cdot E(x, y, t). \tag{3.1}$$

For conventional capture, $S(x, y, t) = 1 \ , \forall(x, y, t)$. Our goal is to reconstruct E from a single captured image I with the control of $S(x, y, t)$.

Equation (3.1) can be written in matrix form as $\mathbf{I} = \mathbf{SE}$, where $\mathbf{I}$ (observation) and $\mathbf{E}$ (unknowns) are vectors with M^2 and NM^2 elements, respectively. Clearly, the number of observations is significantly lower than the number of unknowns, resulting in an under-determined linear system. Using compressive sensing theory, this system can be solved faithfully if the signal $\mathbf{E}$ has a sparse representation $\boldsymbol{\alpha}$ using a dictionary $\mathbf{D}$:

$$\mathbf{E} = \mathbf{D}\boldsymbol{\alpha} = \alpha_1\mathbf{D}_1 + \cdots + \alpha_k\mathbf{D}_k, \tag{3.2}$$

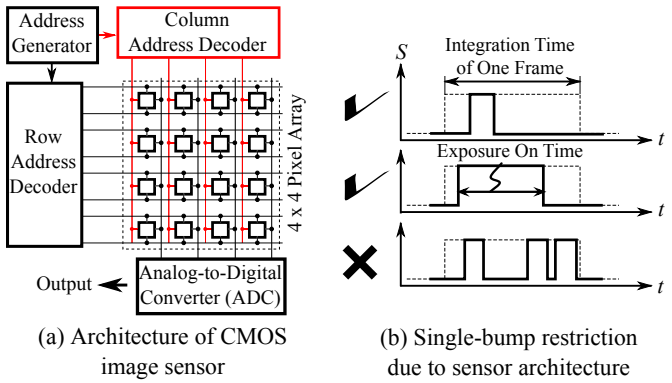
where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_k]^T$ are the coefficients, and $\mathbf{D}_1, \cdots, \mathbf{D}_k$ are the elements in the dictionary $\mathbf{D}$. The coefficient vector $\boldsymbol{\alpha}$ is sparse, which means only a few coefficients are non-zeros. The over-complete dictionary $\mathbf{D}$ is learned from a random collection of videos. At capture time, the space-time volume $\mathbf{E}$ is sampled with a coded exposure function $\mathbf{S}$ and then projected along the time dimension, resulting in a coded exposure image $\mathbf{I}$. Given $\mathbf{D}$, $\mathbf{S}$ and $\mathbf{I}$, $\mathbf{E}$ can be estimated using standard sparse reconstruction techniques.

In the following sections, we will focus on two components of compressive sensing: sampling function (measurement matrix) and representation(dictionary).

3.2 Space-Time Sampling

Most CMOS image sensors have row and column addressing ability (Figure 3.2(a)), and thus it is possible to implement pixel-wise exposure control [2, 49]. However, due to the readout time limit and the fact that most CMOS sensors have no frame buffer on chip, each pixel only allow one continuous exposure during the integration time of one shot (Figure 3.2(b))[1]. For example, assume 0 represents "exposure off" and 1 represents "exposure on", the exposure sequence $[0, 0, 1, 1, 1, 0, 0]$ is realizable while the intermittent exposure sequence $[0, 1, 0, 1, 0, 1, 0]$ is not. Therefore, it is important to adhere to this

[1] CCD image sensors allow multiple bumps (*i.e.*, several individual "exposure on" time within one integration time). However, they usually only have global shutter, and thus do not have per-pixel control.



(a) Architecture of CMOS image sensor

(b) Single-bump restriction due to sensor architecture

Figure 3.2: CMOS sensor architecture and limitations. (a) Current CMOS sensors have row addressing capability (black horizontal connections) which provides row-wise exposure control. Per-pixel exposure control can be implemented by adding column addressing (red vertical connections). (b) Most CMOS sensors do not have per-pixel frame-buffers on chip. Thus, each pixel can have only a single bump (one "exposure on" time) during one camera integration time.

restriction to make our technique implementable on actual CMOS sensors.

We design sampling functions which satisfy the following restrictions imposed by image sensors:

- **Binary shutter:** The sampling function S is binary *i.e.*, $S(x, y, t) \in \{0, 1\}$. At any time t, a pixel is either collecting light (1-on) or not (0-off).
- **Single bump exposure:** Since CMOS sensors do not have per-pixel frame buffers on chip, each pixel can only have one continuous "on" time (*i.e.*, a single bump) during one camera integration time, as shown in Figure 3.2(b).
- **Fixed bump length for all pixels:** Image sensors have a limited dynamic range. A sampling function with a large range of bump lengths among pixels would require a sensor to have a large dynamic range. We consider only the sampling functions with a fixed bump length.

We use the following scheme to assign the bump-start time for all pixels. First, we randomly select the bump-start time of the pixels within a $M \times M$ patch on the top left

Bump length	Noise standard deviation σ (Grey-levels)					
	0	1	4	8	15	40
1	22.96	22.93	22.88	22.50	21.41	17.92
2	23.23	23.22	23.18	23.06	22.62	20.76
3	**23.37**	**23.37**	**23.35**	23.25	23.03	21.69
4	23.29	23.30	23.25	**23.27**	22.99	22.08
5	23.25	23.26	23.24	23.19	**23.07**	**22.34**
6	23.06	23.10	23.07	23.06	22.85	22.32
7	22.93	22.92	22.89	22.85	22.80	22.29
8	22.80	22.81	22.77	22.78	22.69	22.23
9	22.63	22.62	22.61	22.59	22.53	22.09
10	22.49	22.48	22.50	22.49	22.43	22.06

* The highest PSNR value in each column is highlighted in bold.

Table 3.1: Evaluating codes with different bump lengths. For $N = 36$, we generate codes with bump lengths from 1 to 10. For each code, we simulate coded exposure image capture using high-speed video data and add signal-independent noise of varying levels. Peak Signal-to-Noise-Ratio (PSNR) values are computed by comparing the reconstructed space-time volume with the ground-truth.

corner of an image sensor (denoted as p_0), such that the union of the "on" time of these M^2 pixels will cover the entire camera integration time, *i.e.*, $\sum_{(x,y)\in p_0} S(x,y,t) \geq 1$, for $t = 1, \cdots, N$ where N is the number of frames we want to reconstruct from an exposure coded image. Next, consider the adjacent $M \times M$ patch p_1 to the right of p_0. Since there are $M-1$ overlapped columns, we keep the bump-start times for these overlapped pixels, and randomly assign the bump-start times for pixels in the new column in p_1, according to the same constraint for p_0. This process iterates until all pixels have been assigned.

We use simulations to find the optimal bump length. Coded exposure with a long bump length attenuates high frequencies, while coded exposure with a short bump length collects less light, leading to a poor signal-to-noise ratio. For each coded exposure with a given bump length, we simulate coded image capture using real high-speed video data. Signal-independent noise is added to the simulated coded exposure image. From the coded image, we recover the space-time volume using the proposed sparse reconstruction technique. We evaluate peak Signal-to-Noise-Ratio (PSNR) values as a function of the bump length and noise level, averaged over a wide range of scenes. As shown in Table 3.1, as the noise

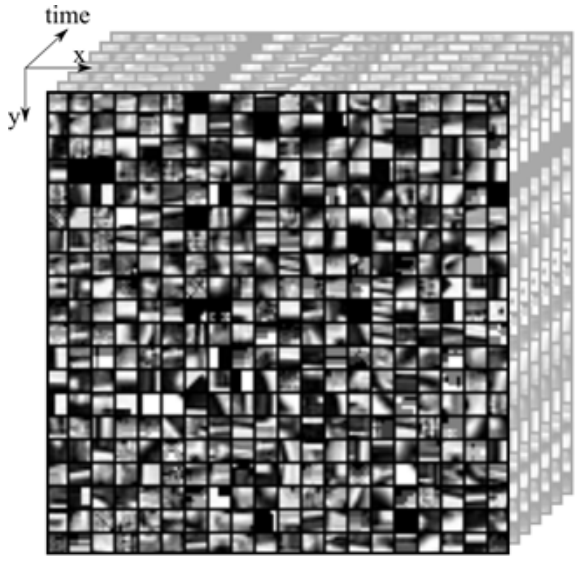


Figure 3.3: Over-complete dictionary learning. Over-complete dictionary is learned from 20 videos of resolution 384×216, rotated into 8 different orientations and played forward and backward. The frame rate of the training videos matches the target frame rate ($500 - 1000$ fps). The learned dictionary captures various local features and structures in videos, such as edges shifting in different orientations.

increases, codes with larger bump lengths are favored. In our experiments, we set the bump length to 2 (for examples with $9X$ gain) or 3 (for examples with $18X$).

3.3 Sparse Representation via Learning

In this section, we discuss the details of building the sparse representation of videos and reconstructing videos from a single exposure coded image. To obtain the sparse representation of videos, we choose to learn an over-complete dictionary from videos covering a wide range of scene, such as racing cars, horse running, skiing, boating and facial expression.

We then model a given video as a *sparse*, *linear* combination of the elements from the learned dictionary (Equation (3.2)). The over-completeness guarantees the sparsity of the representation, and the learning is used to find a dictionary that captures most common structures and features in videos for compact, sparse decomposition.

In our study, we learn an over-complete dictionary on video patches of size $= 7 \times 7 \times 36$, derived from a random selection of videos (20 sequences), using the K-SVD algorithm. The frame rates of the training videos are close to our target frame rate ($500 \sim 1000$ fps). To add variation, we perform rotations on the sequences in eight directions, and play the sequences forward and backward. We learn $5000 \times 20 = 100$K dictionary elements. Figure 3.3 shows a part of the learned dictionary. The dictionary captures features such as shifting edges in various orientations.

Once we learn the over-complete dictionary, we apply a standard sparse estimation technique [13] to recover the space-time volume from a single captured image. Combining Equation (3.1) (for sampling) and Equation (3.2) (for sparse representation), we get $\mathbf{I} = \mathbf{S}\,\mathbf{D}\,\boldsymbol{\alpha}$, where the captured coded image $\mathbf{I}$, the shutter function $\mathbf{S}$, and the over-complete dictionary $\mathbf{D}$ are known. We use OMP to recover a sparse estimate of the vector $\hat{\boldsymbol{\alpha}}$:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\mathbf{SD}\boldsymbol{\alpha} - \mathbf{I}\|_2^2 < \epsilon. \tag{3.3}$$

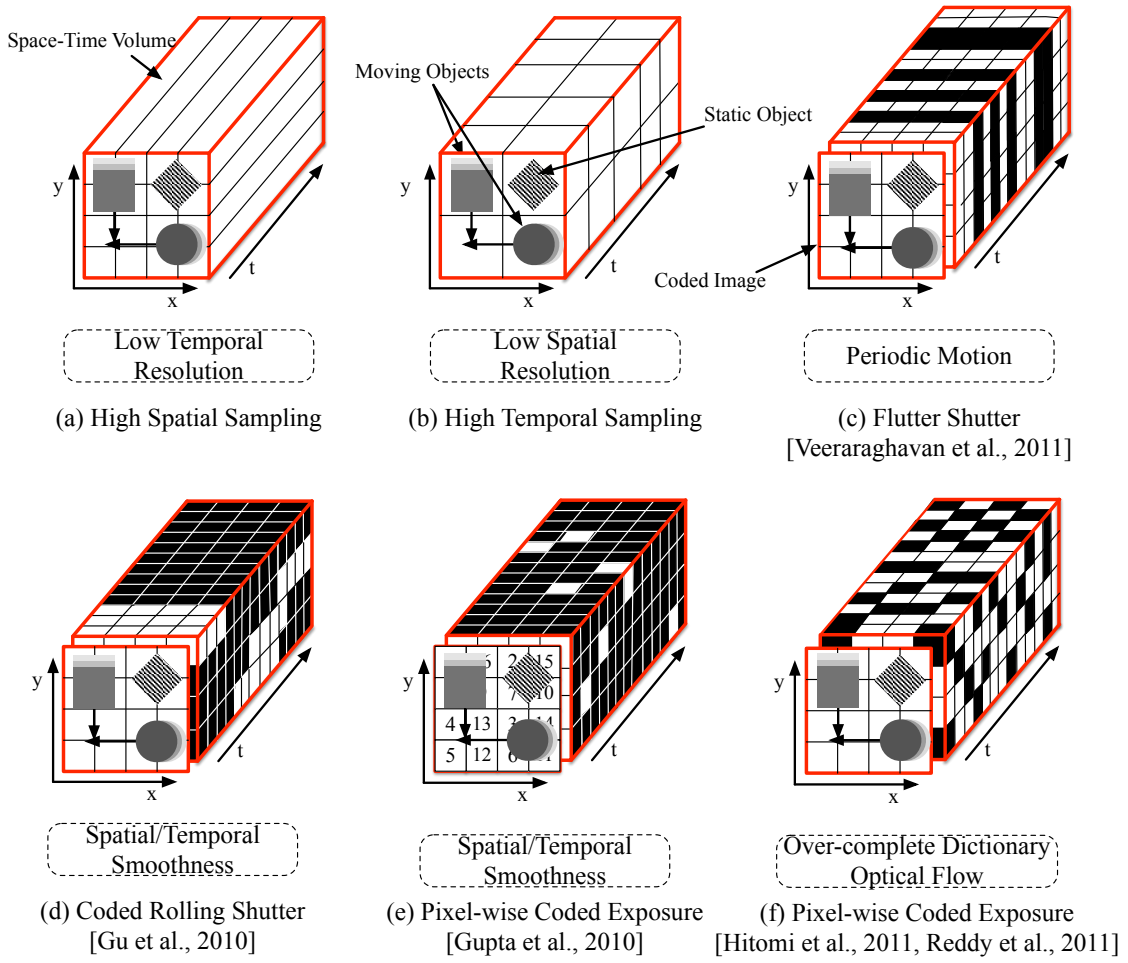
The space-time volume is computed as $\hat{\mathbf{E}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$. We perform the reconstruction for all the $M \times M$ patches in the image. Every pixel (x, y) lies in M^2 patches and thus its time-varying appearance $\mathbf{E}(x, y, \mathbf{t})$ is reconstructed M^2 times. We average these M^2 reconstructions to obtain the final estimate of $\mathbf{E}(x, y, \mathbf{t})$.

3.4 Evaluation and Comparison

In this section, we evaluate the influence factors including sampling function, representation (dictionary), dictionary patch size and noise, which contribute to the final performance of reconstruction.

3.4.1 Sampling Functions

Figure 3.4 shows six sampling functions and their corresponding coded images. Since we are interested in capturing moving scenes, we choose a scene with moving trucks in this figure. Global shutter is the ordinary shutter which exposes the whole image in the integration



(a) High Spatial Sampling

(b) High Temporal Sampling

(c) Flutter Shutter [Veeraraghavan et al., 2011]

(d) Coded Rolling Shutter [Gu et al., 2010]

(e) Pixel-wise Coded Exposure [Gupta et al., 2010]

(f) Pixel-wise Coded Exposure [Hitomi et al., 2011, Reddy et al., 2011]

Figure 3.4: Overview of our work and related space-time sampling schemes. When capturing a space-time volume (the red rectangular box), conventional digital cameras can either have dense spatial sampling with coarse temporal sampling (a) or vice-versa (b). (c) By strobing the exposure, flutter shutter is used to recover periodic motion. (d) Coded rolling shutter is proposed to control the readout timing and exposure length for each row for CMOS sensors. (e) A mixture of denser spatial samples and temporal samples are implemented as a grid shutter for motion-aware high-speed imaging (f) Pixel-wise coded exposure is implemented recently for efficient video capture. A variety of priors and constraints (dashed line boxes in (c)-(f)) are exploited for video reconstruction from a few coded images (red square boxes).

period. As expected, the moving part of the scene is blurred. Flutter shutter [34] opens and closes the shutter many times in an optimized pattern during a single integration time. It preserves some high frequency details, as shown at edges of the moving trucks. Conventional rolling shutter is applied in most CMOS sensors. With the rolling shutter, the whole image is readout row-by-row under the control of row address decoder. One disadvantage of the rolling shutter is the skew effect. Coded rolling shutter [16] is based on the scheme of rolling shutter, but changes the conventional readout sequence, which achieves row-wise exposure control. By using a spatial light modulator, pixel-wise exposure pattern can be implemented. Grid pixel-wise shutter [18] divides the whole image area into several blocks. In each block (*e.g.*, 3×3), an optimized sampling function is applied. Besides a grid exposure pattern, random pixel-wise exposure patterns with a single bump or multiple bumps are also implemented[19, 35]. In order to adhere to the hardware restriction, we choose single bump exposure pattern for comparison.

3.4.2 Sparse Representations

Figure 3.5(a) shows part of the four dictionaries we use for comparison analysis (7×7 patch size). Top left is 3D DCT, the patch on the top left corner is the DC component, thus it only has gray intensity; patches near the bottom right corner represent higher frequencies components. Other patches show patterns with different frequencies. Top right is 3D DWT which is based on Haar wavelets. Bottom left is the learned over-complete dictionary based on 10 different scenes using K-SVD algorithm. Bottom right is 3D random dictionary which is generated based on i.i.d uniformly distributed entries.

Figure 3.5(b) shows the performance comparison for different representations. In this comparison, the same sampling function and reconstruction method are used for all the representations. The comparisons are performed using simulations on high-speed video data. Notice that the learned over-complete dictionary has a higher PSNR as compared to the analytical bases for the same number of bases elements.

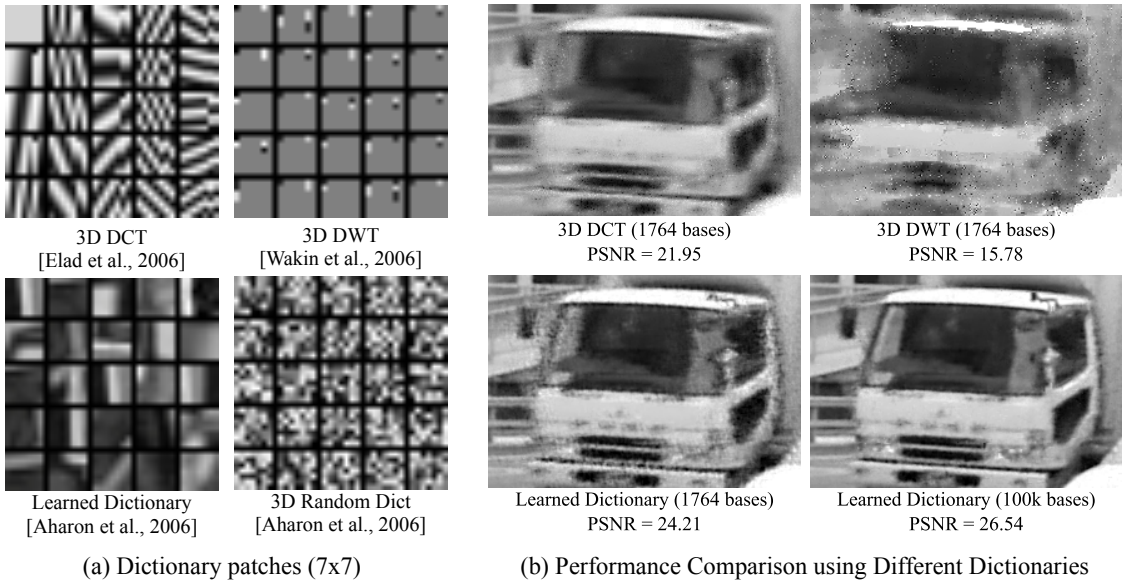


Figure 3.5: (a) Part of four dictionaries (7×7 patch size). The patch on the top left corner of 3D DCT is the DC component, thus it only has gray intensity. When it goes to the bottom right, the frequency of the patch pattern becomes higher. 3D DWT is based on Haar wavelet. Learned over-complete dictionary is trained from 10 different scenes using the K-SVD algorithm. 3D Random dictionary is generated using i.i.d. uniformly distributed entries. (b) Comparison of different representations. Learned dictionaries (bottom row) capture the sparsity in signal more effectively as compared to analytical bases (top row), resulting in better reconstructions. Increasing the number of bases (over-complete dictionary) further improves the reconstruction quality. For this comparison, the same sampling scheme (pixel-wise exposure) and sparse reconstruction algorithm are used.

3.4.3 Coded Sampling vs. Sparse Representation

As shown in the diagram of our approach, both coded sampling function and sparse representation (dictionary) are needed for reconstruction. However, which is more important — coded sampling or sparse representation? To answer this question, we perform a thorough comparison analysis on different combinations of sampling functions and sparse representations.

We select four dictionaries, six sampling functions and five different size of dictionary patches for comparison analysis, which are 120 configurations in total for one scene. All reconstructions are done using the algorithm mentioned in section 3.3. For time efficiency, we use some high performance computing resources from the National Institute for Computational Science (NICS).

Figure 3.6 and Figure 3.7 show the grid reconstruction results for six sampling functions and four dictionaries with 7×7 patch size. The results are the reconstruction of 36 frames from a single coded image. We calculate normalized Root Mean Squared Error (RMSE) and Structural SIMilarity (SSIM) [45]. Notice that the combination of pixel-wise coded exposure and learned dictionary yields the smallest RMSE and the largest SSIM among all configurations. Although the difference in RMSE and SSIM evaluation between grid pixel-wise shutter and random pixel-wise shutter (using learned dictionary) is small, we can still see some difference visually in the reconstruction result. Due to the repetitive structure in grid pixel-wise shutter, there are some jagged artifacts along edges. Whereas, it is smoother in result using random pixel-wise shutter. Besides, coded sampling (either row-wise or pixel-wise) generally results in better reconstruction irrespective of the choice of sparse representation. We run the same simulation on all the test videos in our database and we observe the similar trend. Thus, we conclude that both coded sampling and sparse representation are important to obtain a good reconstruction result, but coded sampling contributes more.

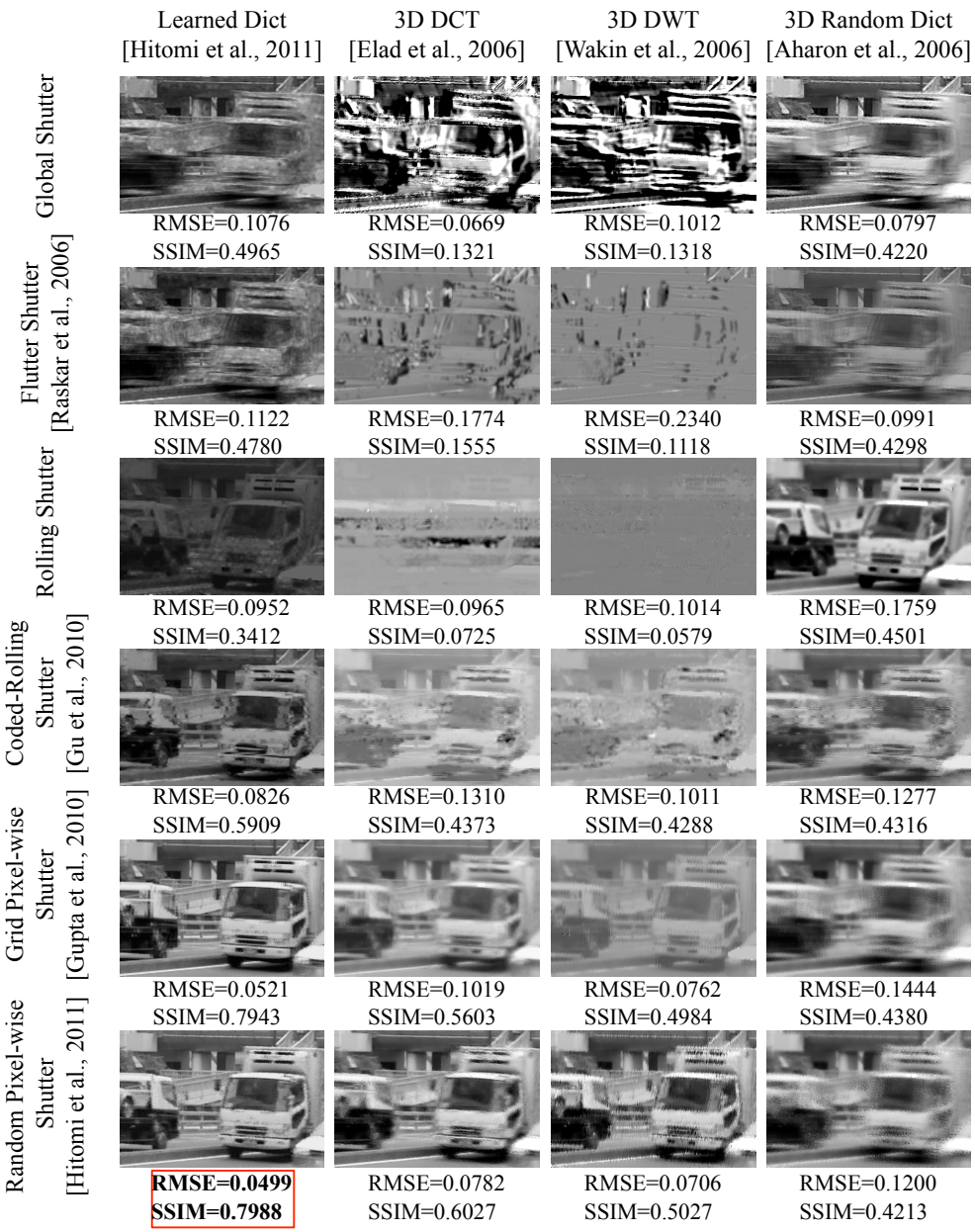


Figure 3.6: Sampling functions versus representations . Horizontal direction shows reconstruction results (36X gain, frame 9 out of 36) for four dictionaries, combined with six exposure patterns along the vertical direction. Numerical analysis is given based on RMSE and SSIM.

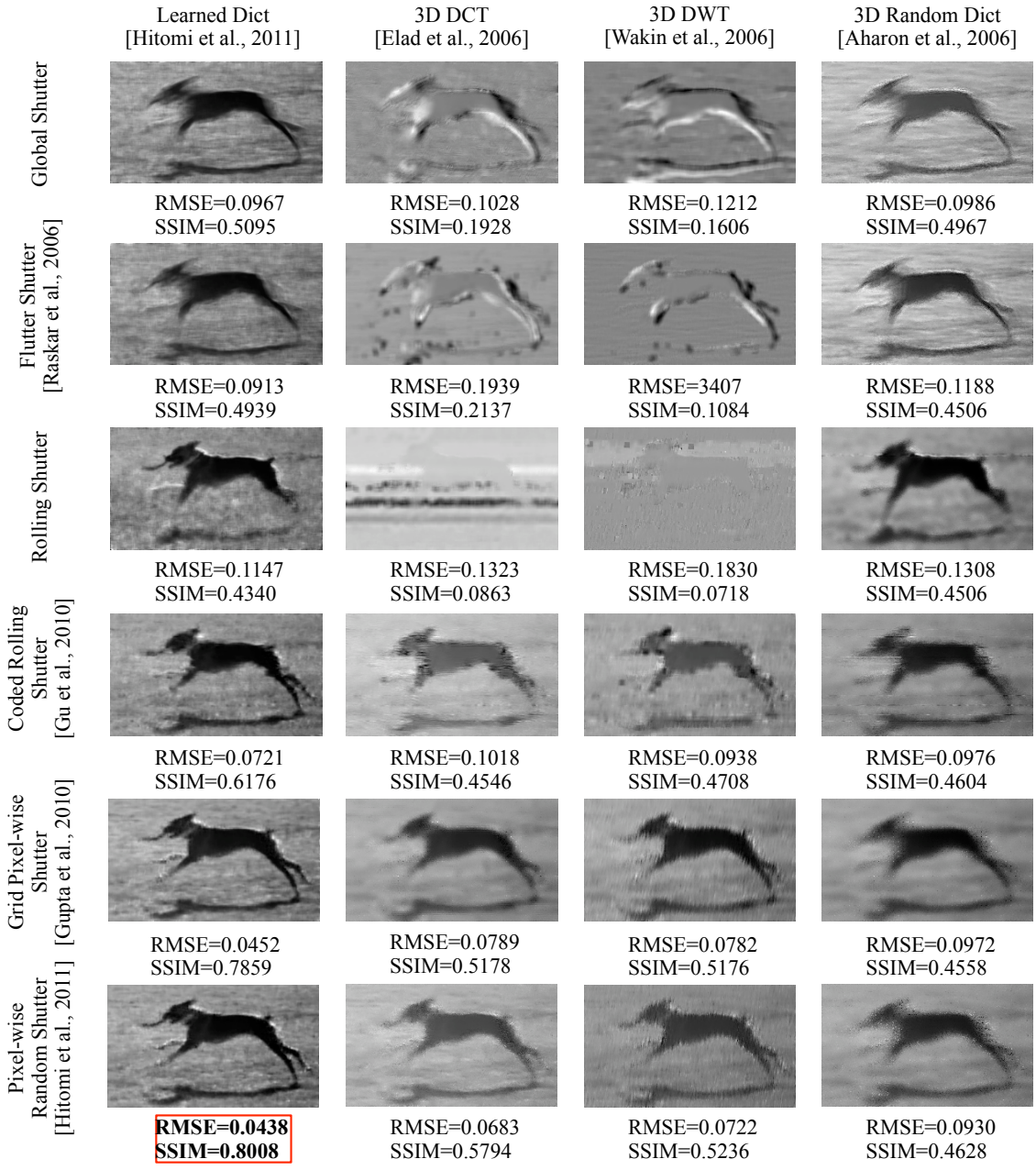


Figure 3.7: Sampling functions versus representations . Horizontal direction shows reconstruction results (36X gain, frame 9 out of 36) for four dictionaries, combined with six exposure patterns along the vertical direction. Numerical analysis is given based on RMSE and SSIM.

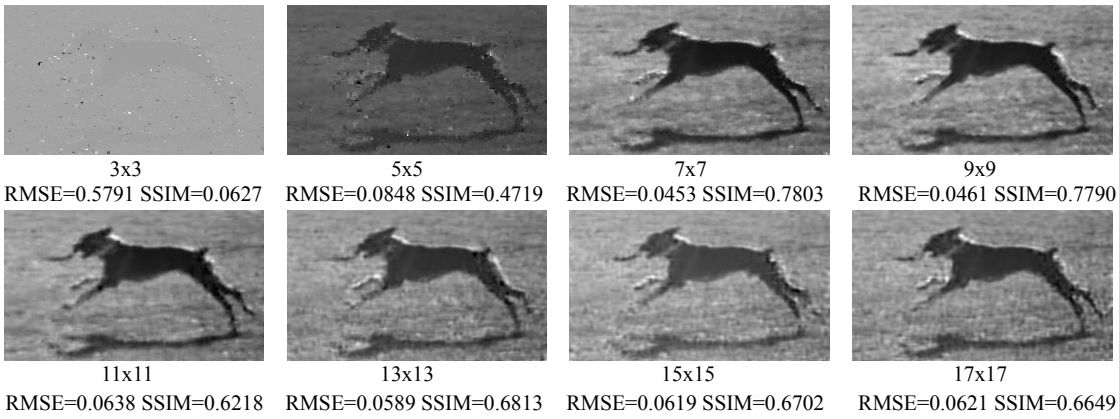


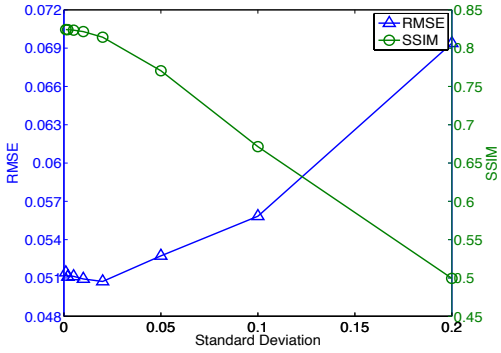
Figure 3.8: Reconstruction results (36X gain) based on eight dictionary patch sizes(showing frame 7 out of 36 video frames). When the patch size is information for the input sources, only gray intensity left, thus the reconstructed result appears gray. When the patch size is too large(*e.g.*, 17×17), the learned dictionary patches only contain general features and lost high frequency information, which can be seen from the grass and dog's back feet.

3.4.4 Dictionary Patch Size

We also analyze the reconstruction results for different dictionary patch sizes using pixel-wise sampling function and learned dictionary, as shown in Figure 3.8. When the dictionary patch size is too small, the learned dictionary patches don't contain any detail information of the source video dataset, but only gray intensity, thus are useless to represent other videos. When the dictionary patch size is too large, it is not efficient to reconstruct some detail features of the scene. As shown in the results for 17×17 dictionary patch, the figure shows the block artifact on dog's legs. At the same time, larger dictionary patch size also requires much longer time to do reconstruction. Considering the performance and time cost, we choose the dictionary patch size as 7×7.

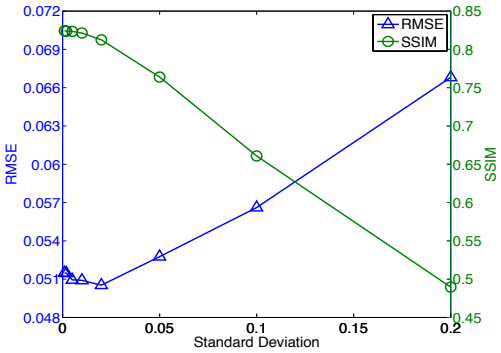
3.4.5 Noise Performance

We simulate reconstruction with photon and readout (Gaussian) noise. Figure 3.9 shows the RMSE and SSIM plot for the truck scene. We evaluate noise performance with mean of the signal power (for photon noise, the square root of signal power), and standard deviation



Reconstructed Video (showing frame 6 out of 36)
Readout Noise Standard Deviation: 0.2

(a) Readout Noise Evaluation



Reconstructed Video (showing frame 6 out of 36)
Photon Noise Standard Deviation: 0.2

(b) Photon Noise Evaluation

Figure 3.9: Noise evaluation of our algorithm: shows RMSE & SSIM curves and two frames of reconstructed video for readout noise and photon noise evaluation. When the standard deviation of noise is 0.2, the RMSE is less than 0.07, and the SSIM is about 0.5.

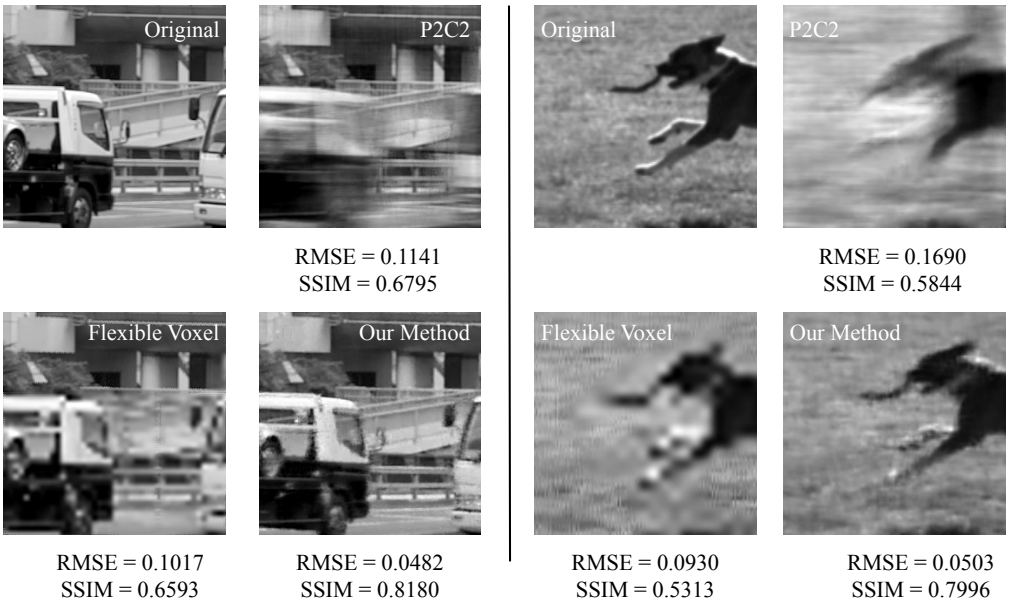


Figure 3.10: Reconstruction results (32X gain) compared with other two methods showing frame 6 out of 32. Compared with other two methods, our method can preserve more features both in background and motion object.

range from 0.001 to 0.2. Two frames of the reconstructed videos are shown with noise of standard deviation of 0.2. The results show that our method is robust to photon noise and readout noise in a relative scale.

3.4.6 Comparison Results with Other Methods

We compare our reconstruction results with recent methods using flexible voxel [18] and P2C2 [35]. We fix the input with only one coded image. Figure 3.10 shows one comparison result for one frame of the reconstructed video of a running dog scene with error evaluation. Flexible voxel method generates different spatial-temporal interpretations from the coded image, and then do motion-aware post-processing interpolation. It preserves high spatial resolution features in static region, but trades off spatial resolution for high speed motion, as we can see the blurry feature on the dog. P2C2 does a good job when using

multiple coded images to calculate optical flow, but if there is only one coded image, the reconstruction result is degraded. In summary, flexible voxel is simple and fast, but limited to simple scenes with few features. P2C2 needs several coded images to better exploit the temporal redundancy. Our method exploits natural video priors by using a dictionary learning based algorithm instead of interpolation or optical flow. Although the time cost is relative high, it outperforms other two methods for most scenarios.

3.5 Conclusion

In this chapter, we demonstrate our method for efficient space-time sampling and reconstruction. Compared to the previous works, our approach has two important distinctions:

- We impose a practical constraint—non-intermittent per pixel exposure—to the sampling function, which makes our approach easier to implement on real image sensors.
- General analytical transforms, such as DCT and DWT, often do not provide the desired level of compactness for sparse representation. Specific motion models, such as periodic motion[43], locally rigid motion[32] and linear motion[38] are only applicable to specific scenarios. In our work, we use a data-driven sparse representation for videos, which is more effective for sparse reconstruction, as shown by experimental results in Section 3.3.

We evaluate the reconstruction performance on different combinations of sampling functions and dictionaries. The results indicate that both sampling functions and sparse representations affect the performance of reconstruction, but sampling functions play a more important role in reconstruction, which give us a hint on the direction of future research.

CHAPTER 4

Hardware Implementation

In this chapter, we give details of our prototype imaging system. We will first briefly describe the Spatial Light Modulator (SLM) and compare three types of popular SLMs. Secondly, we will introduce our prototype camera with per-pixel exposure control. Finally, we will evaluate important system characteristics for our prototype camera.

4.1 Overview of Spatial Light Modulator (SLM)

Our sampling scheme (Section 3.2) requires fast per-pixel modulations. Although we are not able to build a real image sensor with per-pixel exposure control, we can build an emulated imaging system using SLM. SLM is a device that imposes spatially varying modulation on light. SLMs are used extensively in projection display, but they can also be used as a component in optical computing. There are basically three types of SLMs, as shown in Figure 4.1.

Figure 4.1(a) shows the transmissive Liquid Crystal (LC). It modulates the light by changing its polarization state, *i.e.*, when a pixel is turned "ON", S-polarized light will be

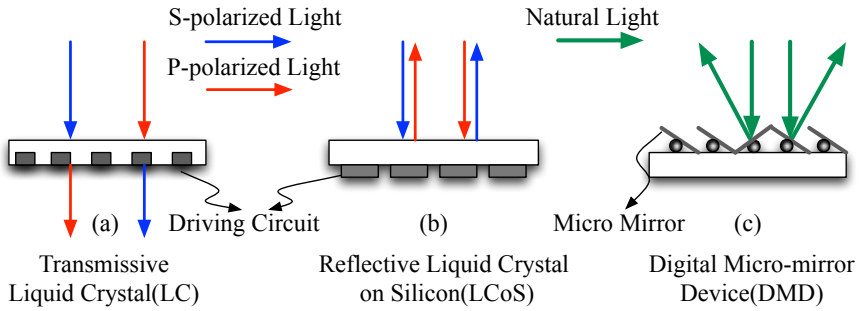


Figure 4.1: Spatial Light Modulator(SLM):(a) and (b) modulate light by changing polarization and pulse width modulation for gray scale operation, (c) modulates light by changing light direction and pulse width modulation for gray scale operation.

changed to a P-polarized light after going through that pixel. Nayar and Branzoi [28] built an adaptive dynamic range imaging system based on LCD. But this kind of device has some limitations. Because the device is transparent, and the driving circuits are located between the liquid crystal elements, this will reduce the fill factor for each pixel. Besides, the pattern generated on the LC is optically defocused by the imaging system and thus pixel-wise control could not be achieved. Finally, due to the diffraction effect produced by the LC cells, the captured images will also be blurred [29].

Another kind of LC device is called Liquid Crystal on Silicon (LCoS), which is a reflective liquid crystal device. Light modulation on this device is also based on changing polarization, but it is reflective. As shown in Figure 4.1(b), the driving circuit is located on the back side of the LC, thus the fill factor and contrast ratio is increased. By locating the LCoS on the virtual sensor plane of the image sensor, pixel-wise control can be achieved in a relative compact imaging system [24, 35].

In order to modulate the light, both transmissive LC and LCoS need a polarizer. A polarizer will reduce the light by half, and combined with other optical components, the light throughput can be greatly reduced [26]. A DMD invented by Texas Instruments (TI) is a Micro-Electro-Mechanical System (MEMS) device that has a tiled micro mirror array, as shown in Figure 4.1(c). Those mirrors can be individually rotated $\pm 10^{\circ}$ to an "ON" or "OFF" state. Therefore, light modulation is implemented by controlling the

Table 4.1: Comparison of SLMs

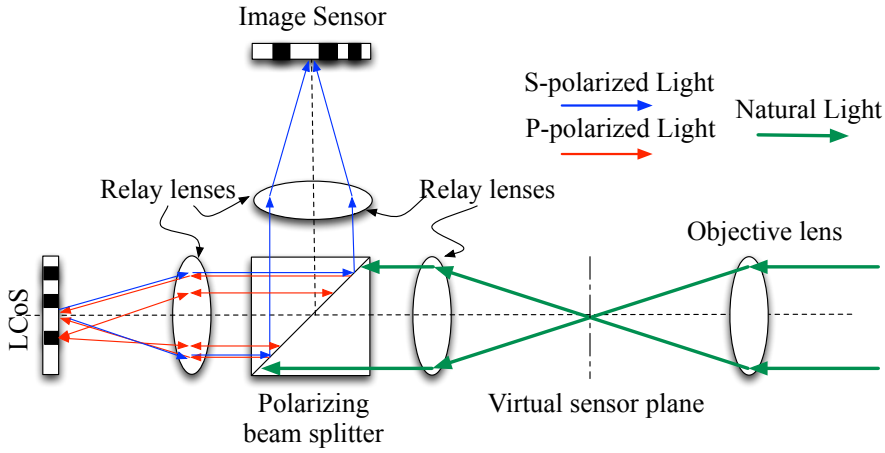
	Transmissive LC	LCoS	DMD
Light Throughput	Low	Medium	High
Frame Rate	Low	High	Medium
Contrast	Low	Medium	Medium
Polarization	Yes	Yes	No
Pixel-wise Control	Difficult	Capable	Capable
Cost	Low	Medium	High

direction of the reflected light from those mirrors. The advantage of using DMD is that no polarizer is needed, and also the reflectivity of DMD mirror is higher than LCoS, so the light throughput of DMD should be higher than LCoS. But since the modulation is achieved by tilting the micro mirror, DMD plane may be not parallel to the image sensor plane, thus lens aberration increases markedly[36].

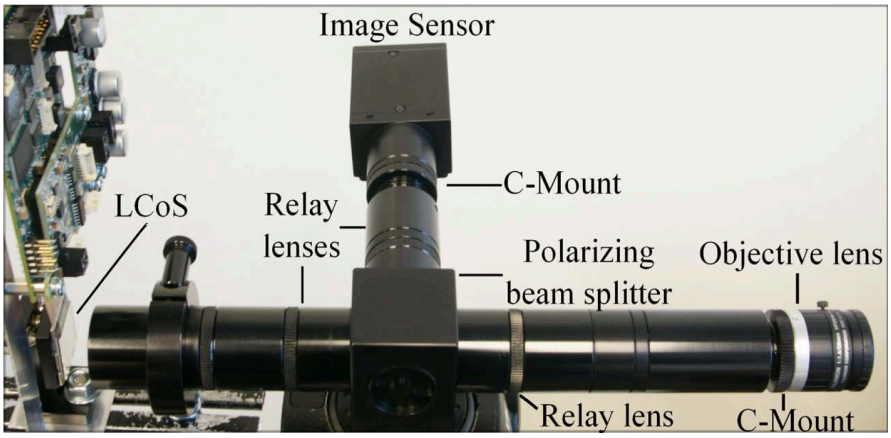
Table 4.1 summarizes these three SLMs in different aspects including the light throughput, frame rate, contrast *etc.*. In general, LCoS and DMD would be a good choice for pixel-wise exposure control.

4.2 Our Prototype

In our prototype, we emulate per-pixel exposure control using a Liquid Crystal on Silicon (LCoS) device. Figure 4.2 illustrates our hardware setup. It consists of an image sensor (Point Grey Grasshopper 2, 1384×1036), an LCoS chip (ForthDD SXGA-3DM, 1280×1024), a polarizing beam-splitter, relay lenses, and an objective lens (Computar M1614 16mm f1.4). The scene is first imaged on a virtual sensor plane through the objective lens, after passing through the polarized beam splitter, S-polarized light is reflected out, and only P-polarized light passes through. Then an image will be focused on the LCoS plane and reflected back. When an LCoS pixel is turned "ON", the P-polarized light will be changed to S-polarized light. When "OFF", the light will be the same (P-polarized). For the reflected light, only S-polarized light will be directed to the image sensor, P-polarized



(a) Optical Diagram of Our Setup



(b) Imaging System Layout

Figure 4.2: Our hardware setup: optical diagram (top) and image (bottom) of our setup. Our system emulates fast per-pixel shutter using LCoS. The incident light is focused after the objective lens, and then becomes collimated after a relay lens and hits the polarizing beam splitter. S-polarized light is reflected away, only P-polarized light passes through. P-polarized light gets focused on LCoS and reflected back. Polarization state of the light changes (S-polarized becomes P-polarized and vice versa) when LCoS pixel is "ON" (shown in white) and keep the same when LCoS pixel is "OFF" (shown in black). At last, only S-polarized light is reflected towards image sensor and P-polarized light passes through.

light will transmit through. Therefore, the incident light is modulated by the LCoS pattern.

The camera and LCoS are synchronized using a trigger signal from the LCoS. During a single camera exposure time, the LCoS displays several binary images, corresponding to the sampling function. We typically run the LCoS at $9 \sim 18$ times the frame-rate corresponding to the integration time of the camera. For example, for an 18ms camera integration time ($55Hz.$), we operate the LCoS at $1000Hz.$, resulting in 18 video frames from a single coded exposure image.

4.3 System Characteristics

4.3.1 Effective F-Number

The F-number is defined as the ratio of focal length to the aperture diameter. F-number affects image depth of field, *i.e.*, photographs taken with a low f-number will tend to have subjects at one distance in focus, with the rest of the image out of focus. The effective F-number of an imaging system is determined by optical components such as relay lenses and the LCoS, not only the objective lens. In our system, after calculation, we find that the relay lens has the smallest F-number which is $f/4$. Therefore, the effective F-number of our imaging system is $f/4$.

4.3.2 Field of View

As shown in the optical diagram in Figure 4.2, the relay system transfers the imaging sensor plane to LCoS plane for light modulation and also to the virtual sensor plane. Since all the relay lenses have the same focal length, the magnification ratio is 1:1. Therefore, the field of view (FOV) of our imaging systems is the same as if the sensor were placed at the virtual sensor plane. The field of view can be calculated using the sensor size and the focal length of the objective lens:

$$FOV \approx 2arctan\frac{d}{2f_o}, \tag{4.1}$$

where d is the diagonal size of the image sensor, and f_o is the focal length of the objective lens.

Our prototype camera use 16mm objective lens and 2/3" ($8.8mm \times 6.6mm$) CCD sensor chip, so the FOV along horizontal and vertical direction can be calculated as 30.75° and 23.31°. We also verify this by taking an image and calculate real FOV based on the distance between objective lens to scene and scene width and height.

4.3.3 Light Efficiency

Light efficiency characterizes how much light is received by the image sensor after passing through the imaging system. Ideally, according to the specification of the LCoS and beam splitter, the light efficiency of the imaging system can be calculated as:

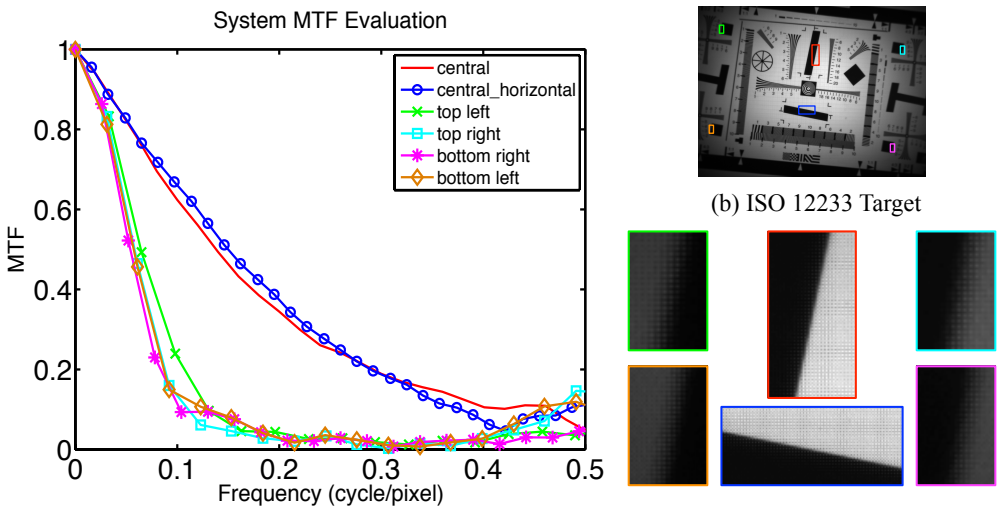
$$27.5\% = 50\%(Polarization) \times 55\%(Reflectivity). \tag{4.2}$$

However, other components such as relay lens may also attenuate the intensity of captured images. Therefore, the actual light efficiency would be lower than 27.5%. To measure the light efficiency of the imaging system, we capture two images of a uniform white scene. One with only objective lens and relay lens, and the other add polarized beam splitter and LCoS. The ratio of the averaged pixel value of those two captured images is calculated as 21.88%, which represents the real light efficiency of the system.

4.3.4 MTF

Modulation Transfer Function (MTF) is one of the most important index for an imaging system. MTF is the spatial frequency response of an imaging system, which describes how well the system is able to resolve image detail as a function of spatial frequency. MTF can be calculated using the following equation:

$$MTF = \frac{M_o}{M_i}, M = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{4.3}$$



(a) System MTF Plot

(b) ISO 12233 Target

(c) Zoom In Slanted Edge Regions

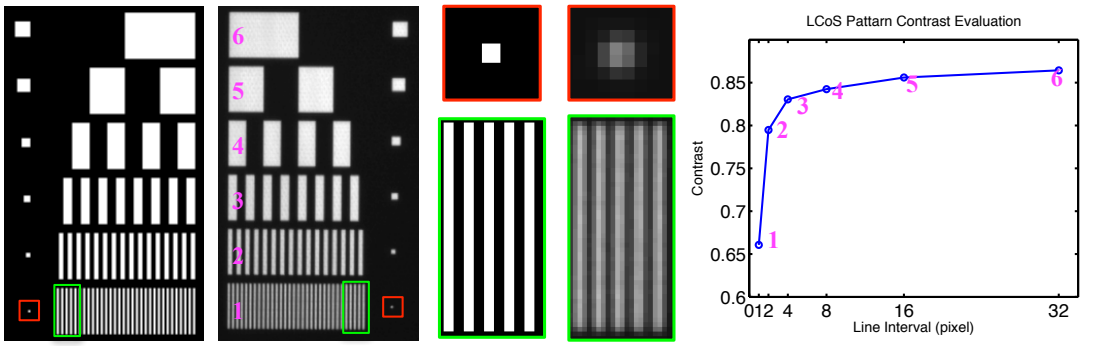
Figure 4.3: MTF evaluation using slanted edge method. (a) MTF curves on different regions of image plane. Central regions of the image plane have higher MTF. (b) ISO 12233 target that we use for measurement. (c) Zoom in of all six regions of edges. The edges in the central part are sharper compared with those in the corner.

where M_o is the output modulation of the image, and M_i is the modulation of the input target.

We evaluate the image plane MTF by capturing a ISO 12233 target image and using slanted edge method [8] to calculate MTF, as shown in Figure 4.3. We select several regions of the image plane to calculate MTF (central region and corner region). The results show that MTF curves in the central region is higher than those in the corner region. The difference is caused by lens aberration. From the zoom in edge regions, we can clearly see that the edges near the corner are blurred. Also notice that the contrast of the edges are decreased.

4.3.5 LCoS Pattern Contrast

In order to evaluate LCoS pattern contrast captured by image sensor, we design a pattern which contains several lines of different frequencies and dots of different sizes, as shown



(a) Ideal Pattern (b) Captured Pattern (c) Zoom in (d) Pattern Contrast vs. Spatial Frequency

Figure 4.4: LCoS pattern contrast evaluation. (a) shows ideal pattern loaded into LCoS, which includes different frequency of line pairs and different size of dots. (b) shows the captured mirror pattern image. (c) shows the zoom in parts of line pairs with single pixel interval and a single pixel dot on ideal pattern and captured image. (d) is the plot of contrast for different frequency of line pairs. X axis shows the line interval in pixels, y axis is contrast and numbers show different regions correspond to the captured pattern.

in Figure 4.4(a). Figure 4.4(b) shows the captured pattern by image sensor, the pattern is mirrored because image sensor captures reflected light from LCoS. From the zoom in regions of ideal pattern and captured image, it shows that the pattern is blurred and the contrast is decreased. That is because in our system, one LCoS pixel corresponds to 2×2 camera pixels. Due to optical blur, a single pixel dot pattern is spread out as shown in Figure 4.4(c). Similar to the MTF evaluation, the contrast decreases as the frequency of LCoS pattern becomes higher.

4.3.6 Vignetting and Distortion

Figure 4.5 shows the evaluation results for vignetting and distortion. Vignetting is evaluated by taking an image of a white scene with uniform illumination. Vignetting is caused by insufficient light coming from the peripheral region. One way to reduce vignetting is to reduce the size of the aperture. The geometric distortion is calibrated by using Matlab camera calibration toolbox.

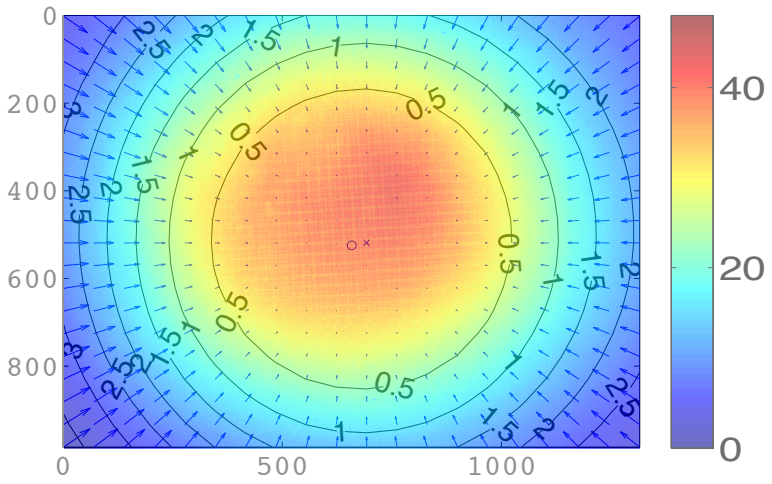


Figure 4.5: Vignetting and distortion evaluation. Captured image of a white scene with uniform illumination shown in false color. Overlapped by distortion model generated by calibration toolbox of Matlab. Number on the contour lines indicates the amount of displacement in pixel unit.

4.4 Experimental Results

Using our hardware prototype, we capture and reconstruct scenes comprising a wide range of motions. Figure 4.6 shows the results. The first example demonstrates the motion of an eye-lid during blinking. This motion is challenging as it involves occlusion and muscle deformations. The input frame is captured with an exposure time of 27ms. Notice the coded motion blur on the input frame. We recover 9 video frames from the captured image, equivalent to an output frame rate of 333 fps.

The second example shows a coin rotating on a table. This motion is challenging due to occlusions; as the coin rotates, one face of the coin becomes visible to the camera. From the single captured image, 9 output frames are reconstructed, while maintaining high spatial resolution, both on the coin and the table. The third and the fourth examples consist of rotating rotor-blades on a toy plane and a ball falling vertically, respectively. The input frames, captured with an exposure time of 18ms show large motion blur. In

order to recover the high-speed motion, we perform the reconstruction at 1000 fps (18 output frames). Notice the sharp edges of the blade and the texture on the ball in the output frames. The spatial detail on the static wings of the toy-plane are nearly the same as the input image. The fifth and sixth examples show the tongue of a flame and the milk drop crown. Note that the subtle change of the flame tongue, as well as the complex fluid motion shown in milk drop, is faithfully reconstructed.

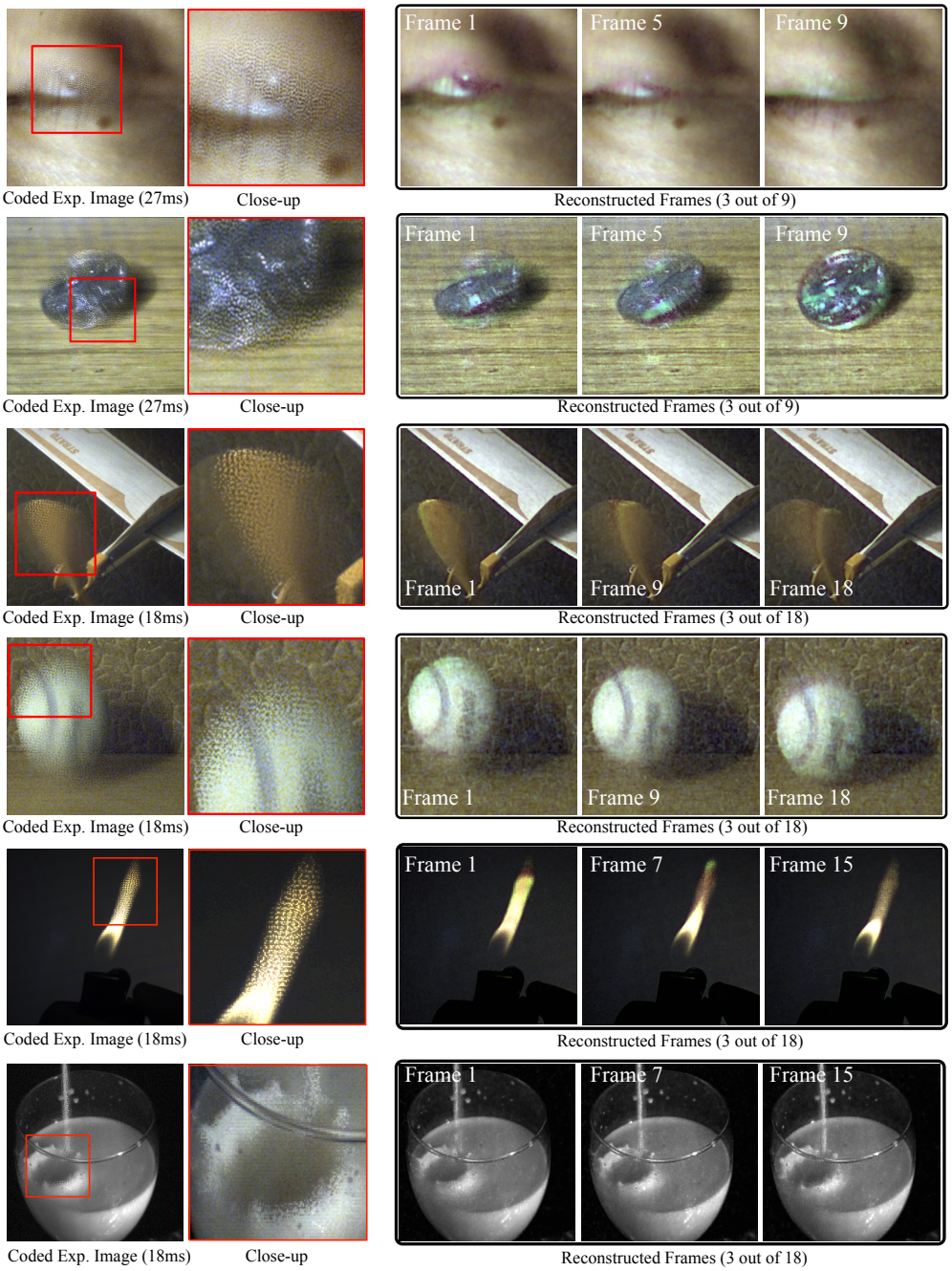


Figure 4.6: Experimental results. **First column:** Input coded exposure images. Numbers in parentheses denote the camera integration time for the input image. **Second column:** Close-ups illustrate the coded motion blur. **Third-sixth columns:** The reconstructions maintain high spatial resolution despite a significant gain in temporal resolution ($9X - 18X$).

CHAPTER 5

Conclusion

Digital camera have revolutionized many fields of imaging, yet due to hardware constraints, there has typically been a tradeoff between the spatial and temporal resolution of these camera systems. By taking advantage of advances in the theory of compressive sensing and the computational power of modern imaging systems, we have been able to obtain more precise control over exposure time, thus achieve flexible space-time sampling. This work should enhance the utility of digital cameras in a wide range of applications including high speed imaging, light field imaging *etc.*.

5.1 Contributions

In this thesis, we propose an efficient way of capturing videos from a single photograph using pixel-wise coded exposure. In summary, there are mainly two contributions:

- We incorporate the hardware restrictions of existing image sensors into the design of sampling schemes, and implement a hardware prototype with an LCoS device that

has pixel-wise exposure control.

- By using an over-complete dictionary learned from a large collection of videos, we achieve sparse representation of space-time volumes for efficient reconstructions. We demonstrate the effectiveness of our method via extensive simulation comparison analysis and experiments.

We aims at capturing videos from a single photograph for a wide range of motions while maintaining high spatial-resolution. Our method does not rely on an analytical motion model, and can handle challenging scenes, including occlusions, deforming objects, flame and fluid flow. Moreover, our sampling function is designed so that it is implementable in real hardware.

5.2 Limitations

There are some limitations both on software and hardware:

Software: First, the temporal resolution of the over-complete dictionary has to be pre-determined (*e.g.*, 36 frames). To do different scales of temporal upsampling, we have to train different dictionaries. Second, the reconstruction time for a video sequence of $450 \times 300 \times 36$ using a 10k dictionary basis is about 5 hours (HP Z600 workstation, 8 cores).

Due to this long running time for training and reconstruction, there are several points that we have not taken into consideration. First, two scenarios are tested under our comparison analysis, and the result supports our conclusions. However, by increasing the test videos in the future, we can strengthen our conclusions. Second, although 20 videos are used as the training database, there is no criterion that can check if they are enough to represent the natural scene. Third, the learning process is restricted to low-dimensional signals, thus learned dictionaries are used on small image patches, rather on the whole image. A 7×7 dictionary patch size is chosen in our project, but an optimal patch size is still desired. Finally, due to the fixed-length of dictionary atoms, only limited temporal

up sampling scales are exploited. There should be a search for an optimal or a flexible dictionary for different scenarios.

Hardware: First, the maximum frame rate of LCoS determines the maximum temporal resolution of the reconstructed high speed video. For example, if the maximum frame rate of LCoS is 1000fps, we can only reconstruct a video of 1000fps at maximum. Second, since the image sensor and the LCoS have different pixel size, one-to-one correspondence requires accurate geometric and radiometric calibration. However, the calibration still has error and can cause artifact (ghosting) and also reduce the contrast of LCoS patterns. We believe that these artifacts can be reduced significantly once the per-pixel exposure control is implemented into the image sensors in the near future.

5.3 Future Work

There are mainly two avenues for future research:

- **Adaptive Exposure**: In this work, the pixel-wise exposure pattern is applied globally to the scene. However, for the static background of the scene, there is no need to apply the randomized exposure pattern. If the exposure patterns can be adaptively changed with respective to the scene (*i.e.*, the exposure time for each pixel can be changed spatial-temporally), there will be mainly two benefits. One is that the SNR for the background can be increased, the other is that the reconstruction time can be reduced since there is no need for reconstruction for the background.

  In order to achieve adaptive exposure, the imaging system should be modified to provide the LCoS with feedback from the captured images. The motion regions from the scene need to be extracted in order to change the exposure pattern adaptively.

- **Adaptive Dictionary**: the learned dictionary is pre-trained and with fixed temporal resolution. Although it is trained from a variety of videos, it is still preferred that the dictionary can be updated adaptively. One possible approach is to adaptively change the elements of the dictionary along with video reconstruction. New dictionary elements can be learned from the reconstructed videos. Another approach

is to build a universal dictionary, where each element is localized in time but can be applied to any instant of the video[30].

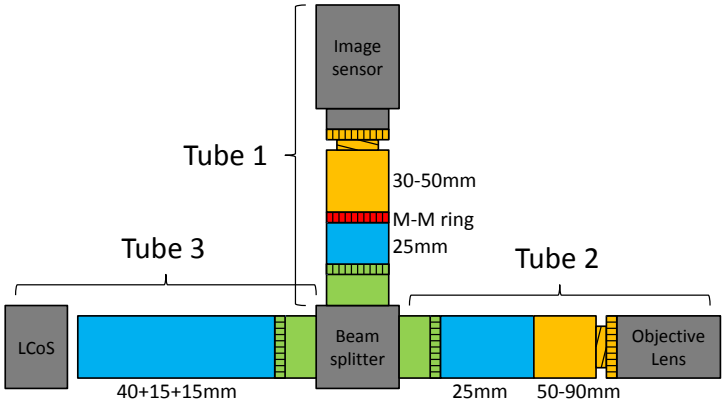
In addition, the added variations on the training video datasets are used to add invariant properties to the learned dictionary, it would be helpful to add verification process to demonstrate the benefit. Besides, techniques such as cross-validation is needed to limit problems like overfitting. Optimization of the dictionary atoms can also be helpful to increase the accuracy of the learned dictionary.

Appendices

APPENDIX A

Optical System Specification

Optical System Specification



Parts List

	Name	Spec.	Stock #	QT
Beam splitter	Beam Splitter	BS CUBE BROADBAND POL VIS 25MM TS	NT49-002	1
	Beam Splitter Case	Mount Only (no optics included) for any 25mm Cube Beam Splitter	NT56-263	1
		C-Mount Male Aperture Cover (Female Threads)	NT58-199	1
		C-Mount Cap (for protecting internal optics or blocking beams)	NT55-245	2
	Extender	C-Mount Extension Tube (5mm Length)	NT54-628	2
		C-Mount Extension Tube (15mm Length)	NT54-630	2
		C-Mount Extension Tube (25mm Length)	NT58-736	2
		C-Mount Extension Tube (40mm Length)	NT54-631	1
	Lens	LENS ACH 25 X 100 MGF2 TS	NT32-327	3
	Lens Mount	C-Mount Achromat/Thick Lens Mount (25/25.4mm Diameter)	NT56-354	3
	Adjuster	C-Mount Fine Focus Tube (30mm - 50mm)	NT03-625	1
		C-Mount Fine Focus Tube (50mm - 90mm)	NT58-757	1
	M-M ring	C-Mount Double Male Thread Ring	NT03-629	1
	LCoS tube connection	C-MOUNT RING MOUNT 1/4-20TAP	NT52-930	2
	LCoS Mount	Metric Mirror Mount- Standard Model, Slotted Base	NT56-342	1

Figure A.1: Optical system specification

References

[1] History of the digital camera and digital imaging. URL http://www.digicammuseum.com/history.html. 1

[2] Sony Develops Next-generation Back-Illuminated CMOS Image Sensor which Embodies the Continuous Evolution of the Cameras. URL http://www.sony.net/SonyInfo/News/Press/201201/12-009E/. 19

[3] Teli Cameras. URL http://www.southimg.com/teli.html. 1

[4] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. G. Narasimhan. Optimal Coded Sampling for Temporal Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 599–606, 2010. 15

[5] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 12, 17

[6] M. Ben-Ezra and S. K. Nayar. Motion-Based Motion Deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689 –698, 2004. 15

[7] G. Bub, M. Tecza, M. Helmes, P. Lee, and P. Kohl. Temporal Pixel Multiplexing for Simultaneous High-Speed, High-Resolution Imaging. *Nature Methods*, 7, 2010. 14

[8] P. D. Burns. Slanted-Edge MTF for Digital Camera and Scanner Analysis. In *IS and TS PICS Conference., Imaging Science & Technology*, page 135, 2000. 40

[9] E. Candès, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. 2

[10] E. J. Candes and T. Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. 8

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998. 10

[12] C. Christopoulos, A. Skodras, T. Ebrahimi, and C. Unit. The {JPEG2000} Still Image Coding Systems: An Overview. *IEEE Trans. Consumer Electronics*, 46(4):1103–1127, 2000. 16

[13] D. L. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52 (4):1289–1306, 2006. 2, 23

[14] M. Elad. *Sparse And Redundant Representations: from Theory to Applications in Signal and Image Processing*. Springer Science & Business Media, 2010. 11

[15] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of Optimal Directions for Frame Design. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2443–2446, 1999. 12

[16] J. Gu, Y. Hitomi, T. Mitsunaga, and S. K. Nayar. Coded Rolling Shutter Photography: Flexible Space-Time Sampling. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2010. 14, 25

[17] A. Gupta, P. Bhat, M. Dontcheva, O. Deussen, B. Curless, and M. Cohen. Enhancing and Experiencing Space-Time Resolution with Videos and Stills. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, 2009. 15, 16

[18] M. Gupta, A. Agrawal, and A. Veeraraghavan. Flexible Voxels for Motion-Aware Videography. In *European Conference on Computer Vision (ECCV)*, volume 3, page 6, 2010. 15, 16, 25, 32

[19] Y. Hitomi, G. J., M. Gupta, T. Mitsunaga, and S. K. Nayar. Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary. In *IEEE International Conference on Computer Vision (ICCV)*, pages 287–294, 2011. 25

[20] J. Holloway, A. C. Sankaranarayanan, A. Veeraraghavan, and S. Tambe. Flutter Shutter Video Camera for Compressive Sensing of Videos. In *IEEE International Conference on Computational Photography (ICCP)*, 2012. 14, 16

[21] S. Kleinfelder, S. Lim, X. Liu, and E. A. Gamal. A 10000 Frames/s CMOS Digital Pixel Sensor. *IEEE Journal of Solid-State Circuits*, 36(12):2049 –2059, 2001. 1

[22] P. Llull, X. Liao, X. Yuan, J. Yang, and D. Kittle. Coded Aperture Compressive Temporal Imaging. *Optics Express*, 58(4):1289–1306, 2013. 14

[23] S. Mallat and Z. Zhang. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 10

[24] H. Mannami, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi. High Dynamic Range Camera using Reflective Liquid Crystal. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 35

[25] R. Marcia, R. Willett, R. Marcia, and R. Willett. Compressive Coded Aperture Video Reconstruction. In *European Signal Processing Conference*, volume 2, 2008. 14, 16

[26] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S. K. Nayar. Programmable Aperture Camera Using LCoS. In *European Conference on Computer Vision (ECCV)*, volume 6316, pages 337–350, 2010. 35

[27] S. K. Nayar. Computational Cameras: Redefining the Image. *IEEE Computer Magazine, Special Issue on Computational Photography*, 39(8):30–38, 2006. 2

[28] S. K. Nayar and V. Branzoi. Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1168–1175, 2003. 35

[29] S. K. Nayar, V. Branzoi, and T. E. Boult. Programmable Imaging: Towards a Flexible Camera. *IEEE International Journal of Computer Vision*, 70(1):7–22, 2006. 5, 14, 35

[30] B. Olshausen. Learning Sparse, Overcomplete Representations of Time-Varying Natural Images. In *International Conference on Image Processing*, volume 1, 2003. 48

[31] B. A. Olshausen and D. J. Field. Natural Image Statistics and Efficient Coding. In *Network: Computation in Neural Systems, 7:333–339*, pages 333–339, 1996. 11, 17

[32] J. Park and M. B. Wakin. A Multiscale Framework for Compressive Sensing of Video. In *Picture Coding Symposium*, pages 1–4, 2009. 16, 33

[33] T. Portz, L. Zhang, and H. Jiang. Random Coded Sampling for High-Speed HDR Video. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, Apr. 2013. 14, 16

[34] R. Raskar, A. Agrawal, and J. Tumblin. Coded Exposure Photography: Motion Deblurring using Fluttered Shutter. In *SIGGRPAH*, volume 3, 2006. 13, 25

[35] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: Programmable Pixel Compressive Camera for High Speed Imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 329–336, 2011. 15, 16, 25, 32, 35

[36] S. Ri, Y. Matsunaga, M. Fujigaki, T. Matui, and Y. Morimoto. Development of DMD Reflection-Type CCD Camera for Phase Analysis and Shape Measurement. *Journal of Robotics and Mechatronics*, 18(6):728, 2006. 14, 36

[37] A. Sankaranarayanan, C. Studer, and R. G. Baraniuk. CS-MUVI: Video Compressive Sensing for Spatial-Multiplexing Cameras. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, 2012. 14, 16

[38] A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, and R. Chellappa. Compressive Acquisition of Dynamic Scenes. In *European Conference on Computer Vision (ECCV)*, volume 6311, pages 129–142, 2010. 16, 33

[39] H. Schaeffer, Y. I. Yang, and S. Osher. Space-Time Regularization for Video Decompression. Technical report, UCLA CAM Report, 2014. 16

[40] E. Shechtman, Y. Caspi, and M. Irani. Space-Time Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(4):531–45, Apr. 2005. 15

[41] X. Shu and N. Ahuja. Imaging Via Three-Dimensional Compressive Sampling (3DCS). In *IEEE International Conference on Computer Vision (ICCV)*, pages 439–446, 2011. 14

[42] Y. W. Tai, H. Du, M. S. Brown, and S. Lin. Correction of Spatially Varying Image and Video Motion Blur Using a Hybrid Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1012 –1028, 2010. 15

[43] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded Strobing Photography: Compressive Sensing of High Speed Periodic Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(4):671–686, 2011. 14, 33

[44] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. Compressive Imaging for Video Representation and Coding. In *Picture Coding Symposium*, 2006. 14, 16

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. In *IEEE Transactions on Image Processing (TIP)*, volume 13, pages 600–612, Apr 2004. 27

[46] G. Warnell, S. Bhattacharya, R. Chellappa, and T. Basar. Adaptive-Rate Compressive Sensing Using Side Information. *arXiv:1401.0583*, 2014. 15

[47] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-Speed Videography using a Dense Camera Array. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 294–301, 2004. 15

[48] J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, and L. Carin. Video Compressive Sensing Using Gaussian Mixture Models. *IEEE Transactions on Image Processing (TIP)*, pages 1–17, July 2014. 15

[49] K. Yonemoto and H. Sumi. A Numerical Analysis of a CMOS Image Sensor with a Simple Fixed-Pattern-Noise-Reduction Technology. *IEEE Transactions on Electron Devices*, 49:746–753, 2002. 19

[50] J. Zheng. Video Compressive Sensing using Spatial Domain Sparsity. *Optical Engineering*, 48(8):087006, Aug. 2009. 16

[51] C. Zhou and S. K. Nayar. Computational Cameras: Convergence of Optics and Processing. *IEEE Transactions on Image Processing (TIP)*, 20(12):3322–3340, 2011. 4, 5